

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.



7.8-10039

NASA CR- 151576
ERIM 122700-31-F

"Made available under NASA sponsorship
in the interest of early and wide dis-
semination of Earth Resources Survey
Program information and without liability
for any use made thereof."

Final Report

PROCEDURE B: A MULTISEGMENT TRAINING SELECTION AND PROPORTION ESTIMATION PROCEDURE FOR PROCESSING LANDSAT AGRICULTURAL DATA

R.J. KAUTH and W. RICHARDSON
Infrared and Optics Division

NOVEMBER 1977

Principal Investigator
Richard F. Nalepka

(E78-10039) PROCEDURE B: A MULTISEGMENT
TRAINING SELECTION AND PROPORTION ESTIMATION
PROCEDURE FOR PROCESSING LANDSAT
AGRICULTURAL DATA Final Report, 15 May 1976
- 14 Nov. 1977 (Environmental Research Inst. G3/43 00039

N78-14456

HC A07/MF A01

Unclas

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Johnson Space Center
Earth Observations Division
Houston, Texas 77058
Contract No. NAS9-14988, Task 1
Technical Monitor: I. Dale Browne/SF3

**ENVIRONMENTAL
RESEARCH INSTITUTE OF MICHIGAN**
FORMERLY WILLOW RUN LABORATORIES, THE UNIVERSITY OF MICHIGAN
BOX 8618 • ANN ARBOR • MICHIGAN 48107



1. Report No. NASA CR- ERIM 122700-31-F		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Procedure B: A Multisegment Training Selection and Proportion Estimation Procedure for Processing Landsat Agricultural Data				5. Report Date November 1977	
				6. Performing Organization Code	
7. Author(s) R. J. Kauth and W. Richardson				8. Performing Organization Report No. 122700-31-F	
9. Performing Organization Name and Address Environmental Research Institute of Michigan Infrared & Optics Division P.O. Box 8618 Ann Arbor, Michigan 48107				10. Work Unit No. Task 1	
				11. Contract or Grant No. NAS9-14988	
12. Sponsoring Agency Name and Address National Aeronautics & Space Administration Johnson Space Center Houston, Texas 77058				13. Type of Report and Period Covered Final Technical Report 15 May 76 - 14 Nov 77	
				14. Sponsoring Agency Code	
15. Supplementary Notes The work was performed for the Earth Observations Division I. Dale Browne (SF3) was the Technical Monitor. Mr. Richard F. Nalepka was ERIM's Principal Investigator. Original photography may be purchased from EROS Data Center Sioux Falls, SD					
16. Abstract The purpose of the work reported here is to develop techniques of training an MSS data classifier suitable for extending signatures over a wide region or partition. The work has the following emphases: a. The effects of sun illumination direction and of haze in the atmosphere are corrected for in order to make data from different times and places comparable. b. Both spectral and spatial data compression are employed in order to concentrate on only the most significant sources of variation. The spectral data compression results in using two linear combinations of the 4-band Landsat data ("brightness" and "greenstuff") which contain most of the information found in agricultural scenes. The spatial data compression is carried out by defining spectrally homogeneous pseudo-fields (called blobs). The total data compression achieved from these steps is approximately a factor of 60. c. Several sites are used simultaneously for training in order to average over unknown sources of site-to-site variation. The central technical issue addressed is how to select the several sites for training. The procedure developed for this purpose is called Procedure B. A proportion estimation technique is included as part of the procedure. An initial demonstration of the procedure using manual training selection methods was moderately successful. Automated training selection techniques were subsequently developed, but the results to date have been less successful than the initial try. Causes of estimation error are being investigated.					
17. Key Words (Suggested by Author(s)) Pattern Recognition Preprocessing Training Methods Atmospheric Effects Signature Extension Remote Sensing Signature Definition Landsat Agricultural Surveys Feature Extraction Crop Identification Multisegment Training				18. Distribution Statement Initial distribution is listed at the end of this document.	
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages	
				22. Price*	

*For sale by the National Technical Information Service, Springfield, Virginia 22161

PREFACE

This report describes part of a comprehensive and continuing program of research concerned with advancing the state-of-the-art in remote sensing of the environment from aircraft and satellites. The research is being carried out for NASA's Lyndon B. Johnson Space Center, Houston, Texas, by the Environmental Research Institute of Michigan (ERIM). The basic objective of this multidisciplinary program is to develop remote sensing as a practical tool to provide the planner and decision-maker with extensive information quickly and economically.

Timely information obtained by remote sensing can be important to such people as the farmer, the city planner, the conservationist, and others concerned with problems such as crop yield and disease, urban land studies and development, water pollution, and forest management. The scope of our program includes:

1. extending the understanding of basic processes.
2. discovering new applications, developing advanced remote-sensing systems, and improving automatic data processing to extract information in a useful form.
3. assisting in data collection, processing, analysis, and ground-truth verification.

The research described herein was performed under NASA Contract NAS9-14988 and covers the period from May 15, 1976 through Nov 14, 1977. I. Dale Browne (SF3) was the NASA Contract Technical Monitor. The program was directed by Richard R. Legault, Vice-President of ERIM and Head of the Infrared and Optics Division, Quentin A. Holmes, Head of the Information & Analysis Department, and Richard F. Nalepka, Principal Investigator and Head of the Multispectral Analysis Section.

The authors acknowledge the contributions of G. Thomas who was a co-innovator of the screening program and who assisted greatly in the development of the overall structure and the initial demonstration of Procedure B.

W. Holsztynski contributed to the training selection procedure and provided mathematical insight into the problems of ancillary variable dependent signatures.

R. Hieber, A. Metzger, S. Lindner, and H. Gallarda assisted in many aspects of programming support and ground truth analysis.

One of the central themes of the procedure we have developed is unsupervised clustering and post-cluster labeling and estimation. This approach, with numerous variations, has come to us from many sources within ERIM and JSC.

CONTENTS

	<u>Page</u>
PREFACE	iii
TABLE OF CONTENTS	v
FIGURES	vii
TABLES	ix
1. INTRODUCTION	1
2. PREPROCESSING/FEATURE EXTRACTION	5
2.1 SUMMARY OF PREPROCESSING STEPS	15
2.1.1 SCREENING	15
2.1.2 DIAGNOSTIC PROCEDURES (INTERIM)	17
2.1.3 CORRECTION PROCEDURES	21
2.2 FEATURE EXTRACTION	23
2.2.1 SPECTRAL FEATURE EXTRACTION	23
2.2.2 SPATIAL FEATURE EXTRACTION	23
3. PROCEDURE B TRAINING AND PROPORTION ESTIMATION	27
3.1 SEGMENT SELECTION	29
3.2 TRAINING FIELD (BLOB) SELECTION	30
4. PROPORTION ESTIMATION	33
4.1 IDENTIFY SELECTED BLOBS	33
4.2 ESTIMATE THE PROPORTION OF WHEAT IN EACH SPECTRAL STRATUM	40
4.3 ESTIMATE SEGMENT PROPORTION	40
5. RESULTS WITH BASELINE PROCEDURE B	43
6. EXTENSION TO THE BASELINE PROCEDURE B	51
6.1 ANCILLARY DATA DEPENDENCE OF SIGNATURES	51
6.2 OPERATIONAL SYSTEM CAPABILITIES	55
6.2.1 THE PROBLEM OF MISSING ACQUISITIONS	55
6.2.2 THE SEQUENTIAL SELECTION PROBLEM	57
7. CONCLUSIONS AND RECOMMENDATIONS	61
APPENDIX 1: A TRANSFORMATION TO MAKE LANDSAT 1 MSS DATA COMPATIBLE TO LANDSAT 2 MSS DATA	63

CONTENTS (CONT.)

	<u>Page</u>
APPENDIX II: PROCEDURE B DATA FLOW	71
APPENDIX II.1: USER DOCUMENTATION OF BLOB	75
APPENDIX II.2: USER DOCUMENTATION OF STRIP	83
APPENDIX II.3: USER DOCUMENTATION OF BCLUST	89
APPENDIX II.4: ANNOTATED LIST OF WCT	95
APPENDIX III: A NOTE ON THE RATIONALE FOR PARTITIONING IN LARGE AREA REMOTE SENSING SURVEYS	101
APPENDIX III.1: HOW TO FIND A METRIC FOR PARTITIONING THE ANCIL- LARY VARIABLE SPACE WHEN THE COVARIANCE MATRIX OF THE ANCILLARY VARIABLE IS SINGULAR	111
APPENDIX III.2: INVARIANCE OF THE CONDITIONAL COVARIANCE	123
APPENDIX IV: LAST MINUTE RESULTS AND AN EXPLANATION OF THE SIGNATURE EXTENSION PROBLEM	137
REFERENCES	147
DISTRIBUTION LIST	149

FIGURES

	<u>Page</u>
1(a). Bands 5 and 6 Are Fairly Uncorrelated	8
1(b). Bands 4 and 5 Are Highly Correlated	8
2. Tasselled Cap	9
3. Cluster Scatter Plots of Several 5x6-Mile Sample Segments	11
4. Schematic Diagram of Tasselled Cap, Showing Place of Water, Clouds, and Cloud Shadows	12
5. Schematic Diagram of Tasselled Cap as Seen with a Standard Haze and as Seen Through Thick Haze	13
6. Graymap of Segment 1852, Third Biowindow, Composite [Brightness/Minus Green] Rule, Designed to Enhance Mile Road Positions	36
7. Stripped Blob Map, Symbols from 1 Through 50	38
8. Stripped Blob Map, Symbols from 0 Through 19	39
9. Estimated Proportion Vs. True Proportion for 17 Blind Sites in Kansas	45
10. Estimates Vs. True for 17 Segments and 6 Training Segments, 69 Spectral Strata	48
11. Estimates Vs. True for a Subset of 10 Sites Chosen to Have Closely Matching Biowindows, 47 Spectral Strata	49

APPENDIX II

1. Data Flow and Programs Comprising Procedure B	72
--	----

APPENDIX IV

1. Portion of a Table of True Percent Wheat (Between Lines) and Number of Pixels (on Lines) by Spectral Stratum and Segment	138
2. Reproduction of Figure 10 (Section 5), Showing Key Segment Numbers	139

FIGURES (CONT.)

	<u>Page</u>
3. Location in Kansas of the Segments Studied, Showing Key Segment Numbers	139
4. Map of Crop Calendar Adjustments (April 18, 1976)	141
5. Typical Early-Season Map of the Crop Moisture Index . . .	142
6. Estimated Vs. True Wheat Percentage for 17 Segments When Longitude is Included in the Stratification	145

TABLES

	<u>Page</u>
1. Coefficients for Feature Extraction, Landsat 2	18,19
2. Procedure B Outline	28
3. Performance of Procedure B on 17 Sample Segments . . .	44
4. Performance Summary for Training and Recognition Segments Considered Separately	46
5. Grouping of Ancillary Variables	52
6. Example Representation of Sub-Strata. Entries in Table Are Number of Blobs Represented by Segment/Spectral Strata Combination	57

APPENDIX I

1. Pass Pairs Used in Landsat 1:Landsat 2 Fitting Procedure	64
2. Diagnostic Data Used in Fitting	65
3. RMS Error of Three Models	68
4. Regression Coefficients for Three Parameter Model C . .	69

INTRODUCTION

The purpose of this work is to develop improved techniques for estimating the amount of an agricultural crop present over a large geographical region or partition. Specifically, the problem is to define a training and classification procedure that will allow valid estimates for the region to be made on the basis of training information obtained from a few segments. Thus the data to be processed is of two types: a small amount of labeled training data from some segments in the region and a large amount of unlabeled data from these and other segments.

Remotely sensed data has three main attributes: spectral, spatial and temporal. The spectral profile for each pixel is provided by the multispectral scanner. The spatial characteristics include the line and point number and the position of the segment in the region. The temporal characteristics include the changes associated with the passage of time during the growing season.

The problem of estimating and classifying in a wide region is complicated by several sources of variation in the data. These include:

- a. systematic external effects, such as haze, viewing angle, sun zenith angle, and scanner calibration which occur independently of which particular crop is in a particular pixel
- b. effects upon particular crops which depend upon ancillary variables such as moisture, growing degree days, crop calendar, etc., which are observable for an entire segment
- c. random noise, due to scanner noise, to within-site variation of an ancillary variable whose site average is known, or finally, to variation in underlying ancillary conditions which are significant in their effects but not being currently observed.

Regarding item c, the noise properties exhibited by multispectral scanner data are generally highly spatially correlated. A simple example

is provided by the fact that the within field variance of the multi-spectral scanner signals is less than the between field variance of multiple examples of the same crop, hence the choice of a reasonable number of pixels from a single field within a segment may constitute insufficient training for that segment. Similarly the choice of a single segment and the fields within that segment as a training data base for a group of segments may constitute insufficient training for that group of segments. For this reason, our development of a training procedure has proceeded on the assumption that multiple segments are needed for training in order to represent the variability of data present in a group of segments.

Some of the data variation can be removed by correcting for known external effects, so that all data are transformed to a standard reference condition. In the procedure herein described, the data are screened so that pixels containing clouds, cloud shadows, bad data and water are detected and flagged, and a haze diagnostic is computed over the array of good data points. Next the data are corrected for differences in satellite calibration (i.e., all Landsat I data is modified to simulate Landsat II data), for sun zenith angle (i.e., all data is made to look as though it was gathered with a sun zenith angle of 39°), and haze (all data is transformed to a standard haze condition).

The noise variation can be lessened and operating efficiency greatly increased by adopting methods of data compression that preserve useful information while averaging out noise. In our procedure we compress the data in two ways, spectrally and spatially.

Spectral compression is accomplished by a linear transformation of the four Landsat bands through a matrix rotation called the Tasseled Cap transform (or the Kauth-Thomas transform) [1]. Most of the significant information regarding agricultural scenes has been found to lie in the plane defined by the first two components of the resulting transformed data; hence, these are retained, and the last two components are discarded, resulting in a data compression of a factor of 2.



Spatial averaging is accomplished by grouping together pixels which are near to each other and are spectrally similar. The spectral average of the groups of pixels, the number of pixels in the group and the average spatial positions of the groups are retained as data features. The groups of pixels are called blobs [4]. As a result of blobbing, the data are further compressed by a factor of 30.

After compressed data features have been extracted, several possibilities exist for making the proportion estimates that establish the crop acreage. They range from classification of the individual feature vectors to maximum likelihood estimation of the proportions to post-classification sampling (bias correction).

Whatever the method of estimation, it is necessary to select training data that adequately represent the range of data variation in the region. For reasons of economy, the training should be restricted to as small a number of segments as possible. Our training procedure, designed to meet these objectives, is to cluster the compressed data features into spectrally similar strata, and then choose a subset of segments that best represents these strata. The training data within the segments are chosen at random subject to the criteria that they represent the strata proportionally and are allocated among the training segments as evenly as possible.

The method of training suggests a simple method of proportion estimation that we include as part of Procedure B. The proportion of wheat in each stratum is estimated directly from the training data and then an average of these stratum proportions, weighted by the number of pixels in each stratum, is calculated to obtain the proportion estimate for the region. This method of proportion estimation is not a necessary part of the procedure. The selection of representative training data could be followed by other, perhaps better, proportion estimation methods.

In summary, the steps of our procedure and the sections of the report describing them are:

- | | |
|--------------------------------|------------------------|
| 1. external effects correction | Section 2.1 |
| 2. spectral data compression | Sections 2.1 and 2.2.1 |
| 3. spatial data compression | Section 2.2.2 |
| 4. selection of training data | Section 3 |
| 5. proportion estimation | Section 4 |

Dependence of the compressed data features upon ancillary data has also been investigated in the effort reported here (Section 6.1). This effort is in a preliminary state of investigation.

Extensions to the procedure to address the problem of missing acquisitions and the problem of predicting segments likely to be helpful in the next biophase are described in Section 6.2.

PREPROCESSING/FEATURE EXTRACTION

The objectives of the preprocessing and feature extraction steps may be several [2]. Their purpose may be to:

- a. Make the data more comprehensible by adjusting all of it to standard conditions of observation.
- b. Eliminate or flag bad or noisy observations in the data.
- c. Make the data more comprehensible by extracting physical features or projecting the data in such a way as to display its physical structure.
- d. Compress the data, retaining most of the information and averaging over noise and redundancy.
- e. Make the distributions of the derived features fit some convenient model such as the multivariate normal distribution.

Several cautions need to be mentioned. First, the above objectives may not all be met by the same preprocessing transformations, in fact in some cases they may be mutually exclusive. Sometimes it may be desirable to carry out transformations on parallel paths ending at different objectives; for example, a linear transformation might be desirable for producing projections which can be examined by a researcher in order to gain insight, whereas a non-linear projection might be used to cause the preprocessed data to fit a normal distribution.

Further, the term "preprocessing" may be somewhat misleading in that it appears to define a computer architecture, in which first the preprocessing steps are performed, then the classification steps, then a proportion estimate is made, etc. In fact, however, all of the different conceptual steps constitute merely one functional relationship between the data and the desired output, and could in practice be carried out in

one step. The preprocessing steps we are defining are conceptual and might be implemented in a variety of architectures.

The organization we have actually used is as follows: first the data is screened and diagnostic characteristics are extracted from the data. Second, corrections are made to the data and spectral features are extracted. Third, spatial features are extracted.

Finally, the process of defining feature extraction algorithms is itself largely outside the field of pattern recognition. "In light of all the possibilities mentioned above, what feature extraction steps might help solve the problem at hand?" Such a formulation might be posed to a man, but not to a machine, at least not yet. The features extracted are based on a gestalt, on an overall conceptual basis of what the key problem elements are. We proceed now to outline the conceptual basis for the preprocessing and feature extraction steps described in this report.

The Conceptual Basis

In order to decide what features to extract, one first of all needs some fairly accurate global picture of the data structure. We will concentrate first on the spectral structure of the Landsat data from an agricultural scene, with some reference to the temporal structure.

The Tasselled Cap

The Tasselled Cap [1],[3], is a visual-graphic presentation of the structure of Landsat data in agricultural scenes. The Landsat MSS gathers data in four somewhat overlapping spectral bands, Band 4 through Band 7. The data acquired by Landsat is however quite spectrally correlated so that in fact almost all the data lies on a 2-dimensional plane in the four dimensional space. Further, the data is confined to a triangular (Tasselled Cap) shaped region of that plane.

Figures 1 and 2 describe these ideas in more detail. Figure 1(a) shows an artistic conception of a scatter plot of Band 5 vs. Band 6. Band 5 is centered in the chlorophyll absorption band of green vegetation around $0.65 \mu\text{m}$ whereas Band 6 is centered on the cellulose reflectance peak of green vegetation around $0.75 \mu\text{m}$. The signal from green vegetation is thus found to be small in Band 5 and large in Band 6. Such a point is indicated on Figure 1(a) by the designation "green stuff".

The main variability found in soils is in their brightness. Hence, the signals from bare soils are distributed primarily along a line radiating from the origin.

Band 4 is centered on the cellulose reflectance around $0.55 \mu\text{m}$ but extends significantly into the chlorophyll absorption region, so that Bands 4 and 5 are highly correlated as shown in Figure 1(b).

Figure 2 expands the description to three dimensions. Soil sample points fall near a line and, further, fall predominantly in a planar region surrounding that line. Plants start out on bare soil and grow towards the region of green stuff. Among the variety of green plant canopies, some have large amounts of shadow, which shifts the observation directly toward the origin. Trees are an example of a green canopy with a fair amount of shadow, as shown by the "badge of trees". The variety of shadowing in various canopies creates a region called the green arm.

When a particular plant canopy has reached its own appropriate stage of maximum green development it may then yellow. Often yellowing is accompanied by darkening, either in the vegetation itself or due to the drooping of its leaves which causes shadowing to increase. Either by yellowing, withering, or by being cut down, every canopy eventually returns to the soil from whence it came.

In Landsat data the yellow point is so close to the "side" of the tasselled cap that it is almost indistinguishable. Hence, for most purposes we may say that the agricultural information is substantially contained in a plane defined by unit vectors in the "brightness" and

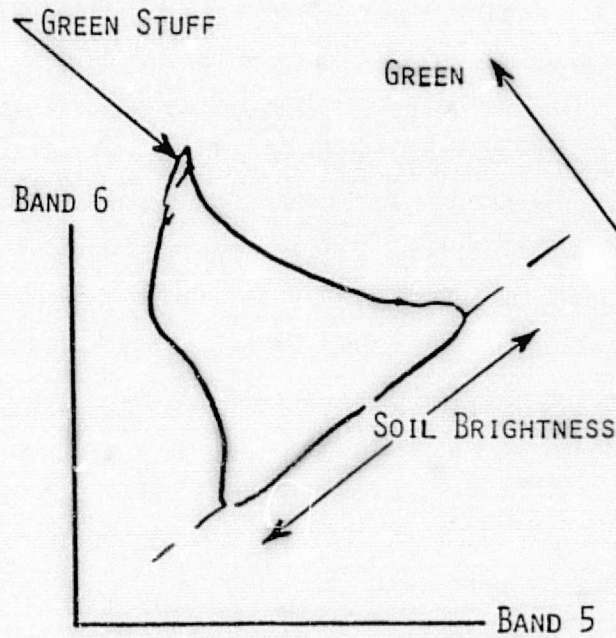


FIGURE 1(a). BANDS 5 AND 6 ARE FAIRLY UNCORRELATED

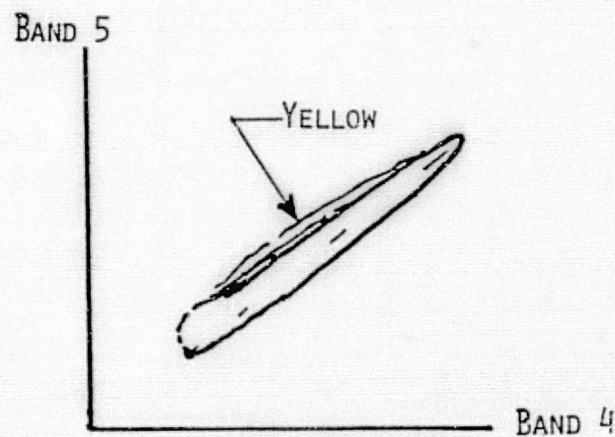


FIGURE 1(b). BANDS 4 AND 5 ARE HIGHLY CORRELATED

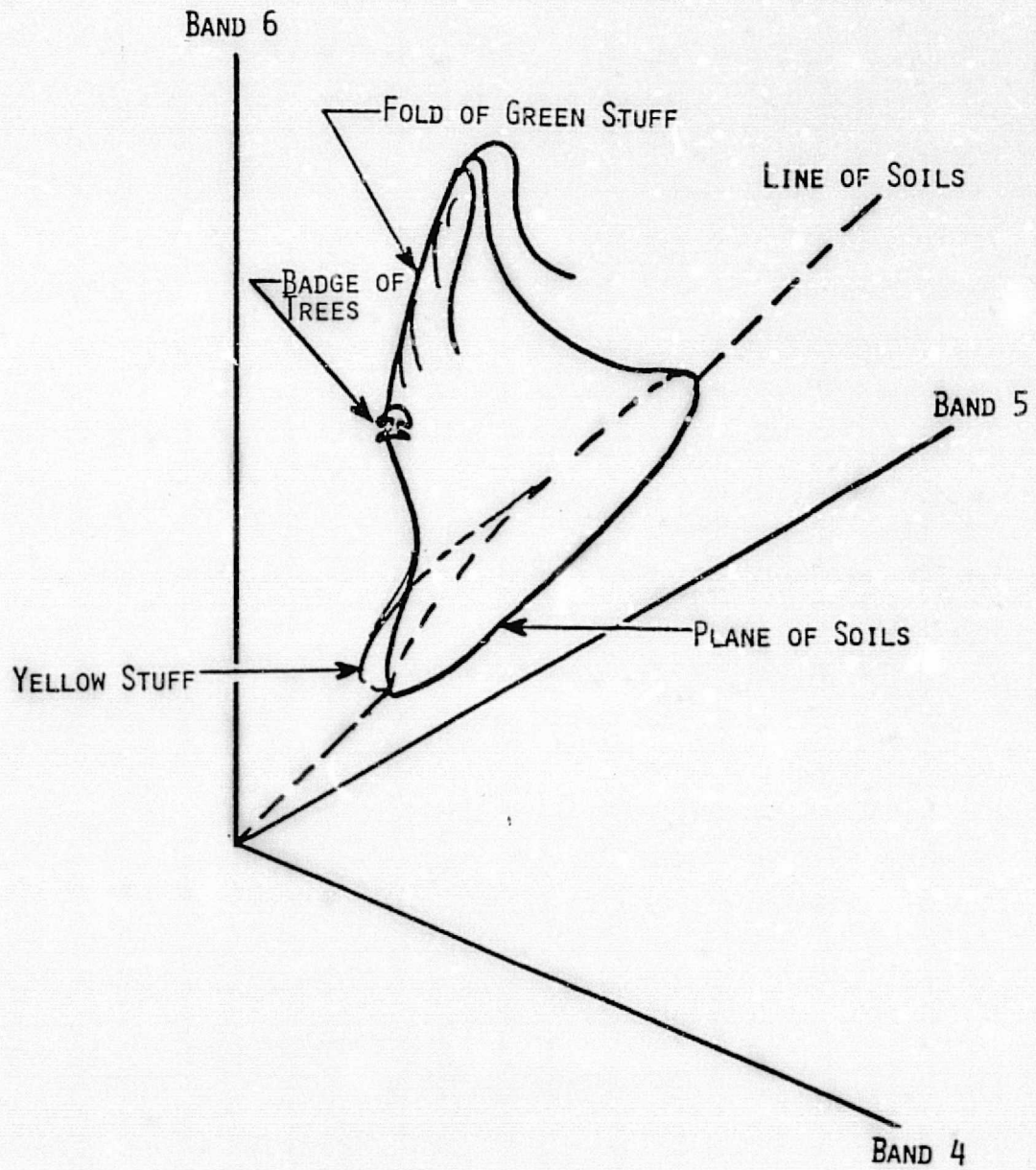


FIGURE 2. THE TASSELLED CAP

"green stuff" directions, but with a small amount of variation in the "yellow stuff" direction perpendicular to the plane. A fourth direction "non-such", orthogonal to the other three, contains primarily noise variation.

Figure 3 shows some typical cluster plots of Landsat data from 5 x 6 mile sample segments. In these figures the abscissa is "brightness" and the ordinate is "green stuff". Notice that, in any particular scene, portions of the Tasselled Cap structure may be missing, depending on the crops planted and their state of development and on the types of soil. (The numbers on the ellipses are only for identification.)

Additional Structural Characteristics of the Data

Signals from vegetation and soil form the Tasselled Cap. Water, clouds, cloud shadow, and haze also have their proper places in the four dimensions of Landsat signal space, and can be described relative to the position of the Tasselled Cap and the points already defined, as shown in Figure 4.

Haze Correction

Figure 4 shows that clouds are located out along the brightness axis, but shifted in the negative yellow direction as well. Haze can be thought of as intermediate or thin cloud. When haze is present over a scene the contrast of the scene itself is reduced while a portion of a cloud-like signal is added. The intermediate condition between no haze and completely hazy (i.e., cloud) is sketched in Figure 5 as a shaded region. We see that as the haze amount is increased the triangular shape of the tasselled cap shrinks towards the point of clouds.

This demonstrates why haze has a severe effect upon signature extension capability. The entire data structure shifts out along the brightness axis and shrinks in the green direction, but worse, the shift in the negative yellow direction is sufficient to move the Tasselled Cap sidewise completely off of its haze free image.

If the data points are first projected onto the two-dimensional brightness-green plane the problem becomes less severe. Even then, however, there will be significant differences between hazy and haze

IN TRANSFORMED COORDINATES

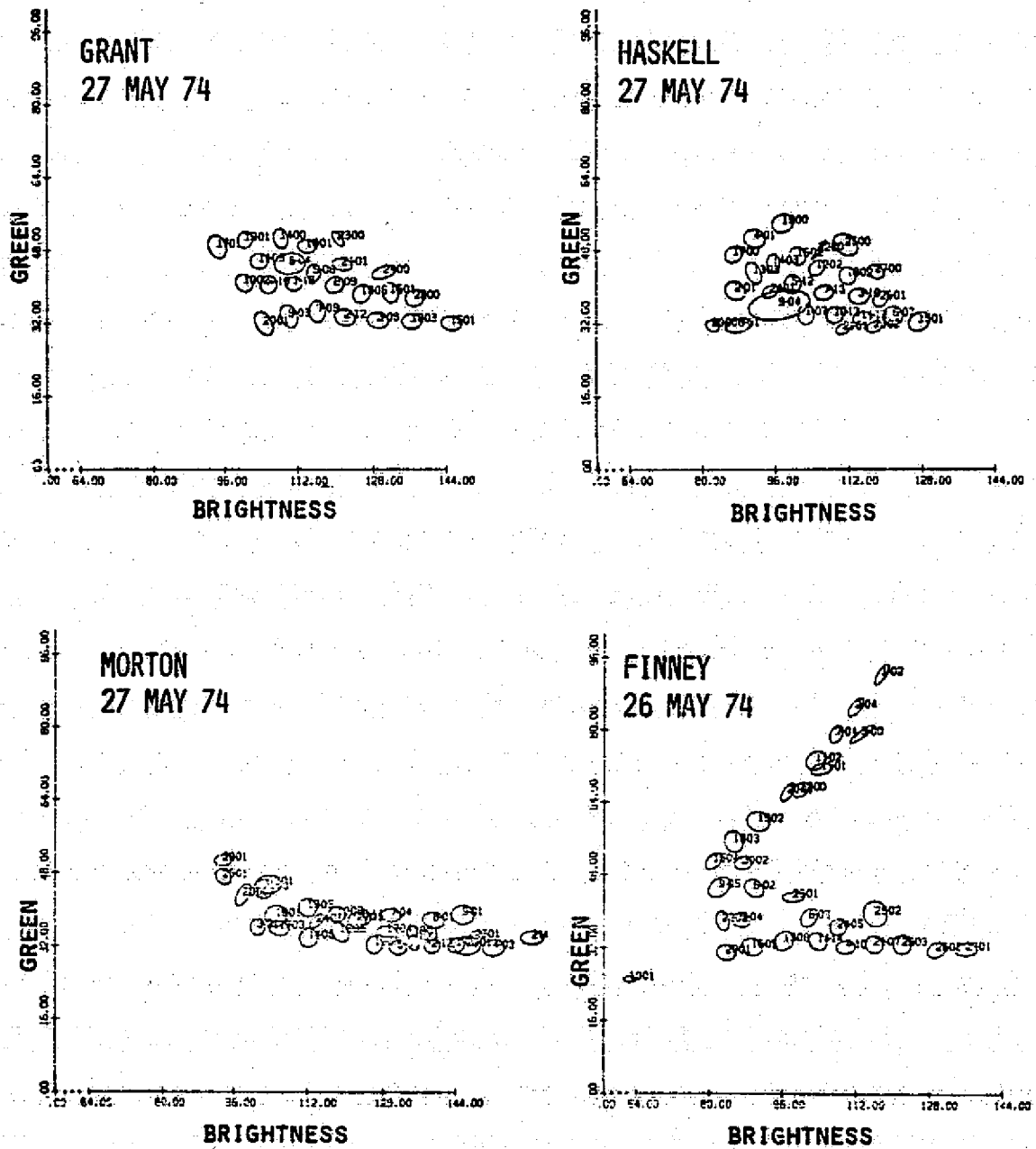
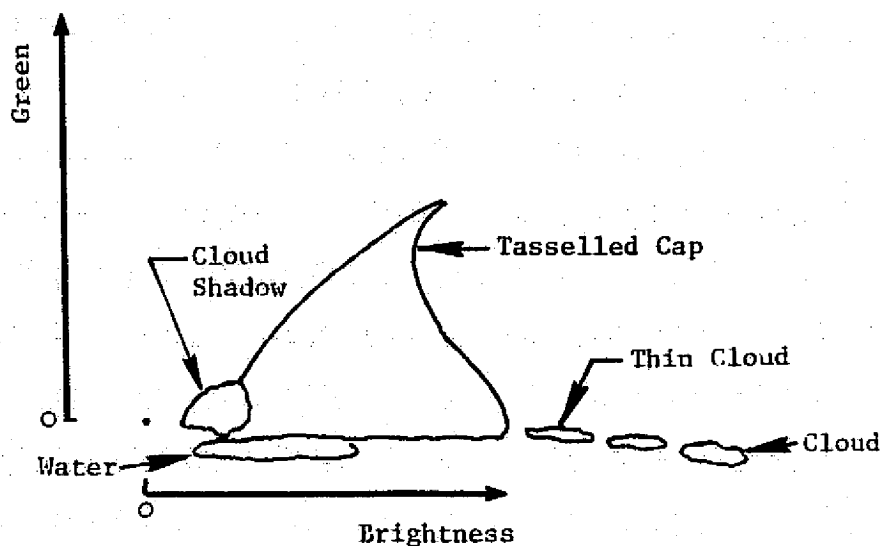
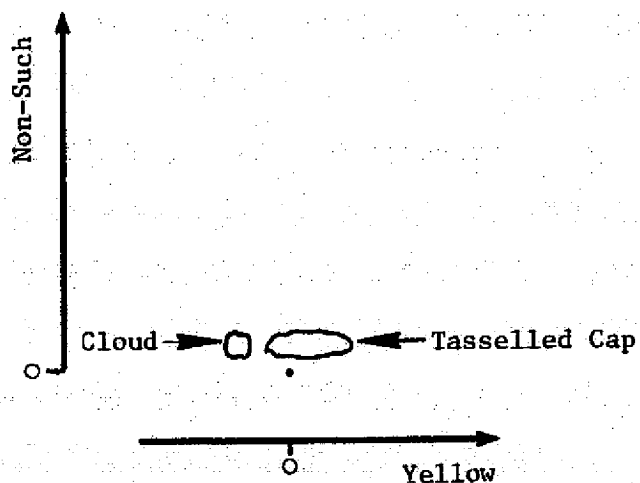


FIGURE 3. CLUSTER SCATTER PLOTS OF SEVERAL 5x6-MILE SAMPLE SEGMENTS



(a) Projection on Bright/Green Plane



(b) Projection on Yellow/Non-Such Plane

FIGURE 4. SCHEMATIC DIAGRAM OF THE TASSELLED CAP, SHOWING THE PLACE OF WATER, CLOUDS, AND CLOUD SHADOWS

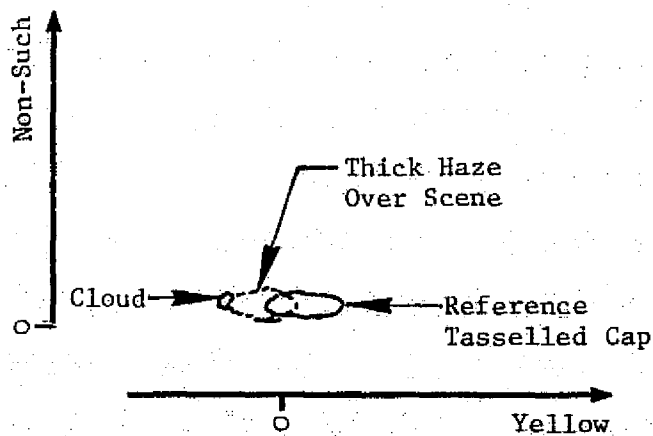
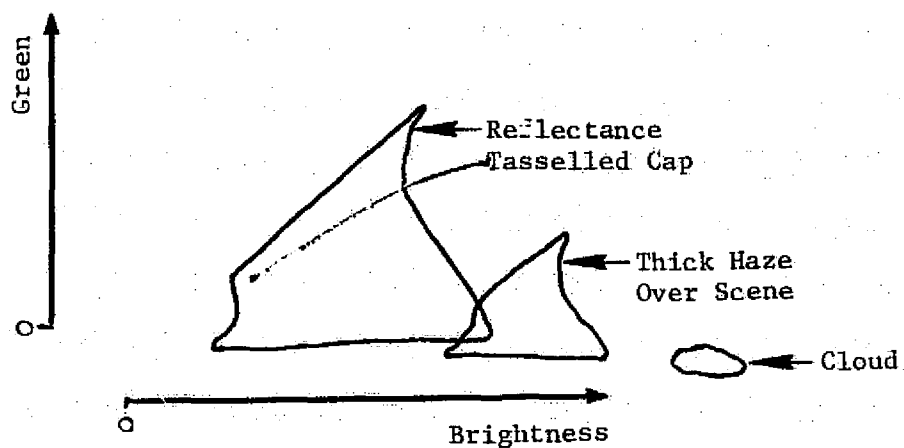


FIGURE 5. SCHEMATIC DIAGRAM OF TASSELLED CAP AS SEEN WITH A STANDARD HAZE AND AS SEEN THROUGH THICK HAZE

free signatures. The proper thing to do is to measure the amount of haze present and adjust the data back to some reference condition.

Fortunately, for the data we have examined to date, the measurement of the amount of haze is possible by measuring the shift of the data in the negative yellow direction.

In order to actually correct the data an atmospheric model which describes the effect of haze is needed. The entire procedure of making the haze measurement and applying the atmosphere model was developed by P. Lambeck of ERIM and is called the XSTAR algorithm. It is described in detail in Reference 5. It will be described in summary form in a later section in this report.

The Effects Of Sun Zenith Angle

To a first order of correction the radiance incident at the Landsat detector system decreases directly with the cosine of the zenith angle of the sun at the earth point viewed. (This would be exactly so if the atmosphere haze scattering and the ground reflectance both behaved as Lambertian reflectors.) In order to make the data gathered under different sun zenith angles commensurate, it is all corrected to the standard zenith angle of 39° which is typical of Landsat data over Kansas in April. This is done before haze correction so that the XSTAR algorithm can operate on a more standardized set of data. The form of the corrected data is

$$X' = \frac{\cos \theta_0}{\cos \theta} X$$

where

θ is the sun zenith angle and $\theta_0 = 39^\circ$, X is the data vector and X' is the data vector corrected for sun angle.

Correction For Satellite Calibration

During the course of this work it was found useful to incorporate data from both Landsat 1 and Landsat 2 passes, although Landsat 2 was

the primary available satellite. The two satellite sensors have slightly different calibrations, which would hardly be noticeable except when attempting machine processing. Hence, it was necessary to find a transformation which would convert the Landsat 1 data to be commensurate with the Landsat 2 data. The procedure by which this was accomplished is detailed in Appendix I. Briefly, it consists of comparing pairs of Landsat 1 and Landsat 2 observations on the same sample segments on successive (9 day separated) passes.

In order to actually carry out the fit it was necessary to take account of the differing haze levels and differing sun zenith angles for the two observations, and to, in effect, assume a non-linear model for each of the satellites (although a useful linear relationship between the two was found).

2.1 SUMMARY OF PREPROCESSING STEPS

We have now discussed most aspects of the preprocessing in a conceptual way. The actual corrections were implemented in a two-step process. In the first step the raw data was "screened" and haze diagnostics were calculated. In the second step the haze correction and all other corrections were applied to the data. (In the same step spectral features were extracted, as will be described in Section 2.2.1.)

2.1.1 SCREENING*

The various data screening procedures are described in this section. During this process each acquisition is screened and each pixel is flagged as to its status:

Bad Data

Cloud

* The screening parameters shown in this section are the ones that were actually used in the work described in this report. Subsequently, an improved version of the screening program, incorporating revised definitions of the decision surfaces, has been developed by P. Lambeck and is documented in Reference 5. If this data were to be reprocessed today, the improved version would be used. The conceptual basis of the screening program is the same in both versions.

Water

Cloud Shadow

Good Pixel.

In the following, R's are four component Tasselled Cap unit vectors* and C's are scalars. It is assumed that the data has been converted to be commensurate with Landsat 2. Table 1 lists the values for these coefficients as well as all others used in this section.

Bad Data

If any of the following conditions are true, the pixel, x, is labeled bad data.

$$Z_4 \stackrel{\text{def}}{=} \left(\frac{\cos \theta_0}{\cos \theta} \right) R_4^T X > C_4$$

$$Z_4 \stackrel{\text{def}}{=} \left(\frac{\cos \theta_0}{\cos \theta} \right) R_4^T X < C_4'$$

$$Z_3 \stackrel{\text{def}}{=} \left(\frac{\cos \theta_0}{\cos \theta} \right) R_3^T X > C_3$$

$$Z_3 \stackrel{\text{def}}{=} \left(\frac{\cos \theta_0}{\cos \theta} \right) R_3^T X < C_3'$$

$$Z_2 \stackrel{\text{def}}{=} \left(\frac{\cos \theta_0}{\cos \theta} \right) R_2^T X < C_2'$$

$$\begin{aligned} Z_L &\stackrel{\text{def}}{=} \left(\frac{\cos \theta_0}{\cos \theta} \right) (.73) R_1^T X + \left(\frac{\cos \theta_0}{\cos \theta} \right) (.68) R_2^T X \\ &= .73Z_1 + .68Z_2 > C_{12} \end{aligned}$$

* The Tasselled Cap transform used here was developed by P. Lambeck for use on Landsat 2 data. The original transform, Reference 1, was developed for Landsat 1.

$$Z_A \stackrel{\text{def}}{=} .73Z_2 - .68Z_1 > C_{21}$$

where

$$Z_1 \stackrel{\text{def}}{=} \begin{pmatrix} \cos \theta \\ \cos \theta \end{pmatrix}_0 R_1^T X$$

The above tests effectively enclose the Tasselled Cap, cloud, water, and cloud shadow. Most bad data points will be excluded by this test, since the volume enclosed is only a small fraction of the total volume of the Landsat signal space.

Cloud

If not bad data and if the following is true, the pixel is labeled cloud.

$$Z_1 - Z_3 > C_C$$

Water

If not bad data or cloud and if the following is true, the pixel is labeled water.

$$Z_1 < C_{w1} \text{ and } Z_2 < C_{w2}$$

Cloud Shadow

If not any of the above and if the following is true, the pixel is labeled cloud shadow.

$$Z_1 < C_S$$

Good Pixel

If none of the above, the pixel is labeled good.

2.1.2 DIAGNOSTIC PROCEDURES (INTERIM)

Diagnostic procedures consist of measuring the data average, the soil mean, and the green arm mean.

TABLE 1. COEFFICIENTS FOR FEATURE EXTRACTION. LANDSAT 2

<u>Name</u>	<u>Symbol</u>	<u>Landsat 2 Band</u>	<u>Coefficient Value</u>
Reference solar zenith angle	θ_0		39°
Brightness unit vector	R_1	4	0.33231
		5	0.60316
		6	0.67581
		7	0.26278
Greenstuff unit vector	R_2	4	-0.28317
		5	-0.66006
		6	0.57735
		7	0.38833
Yellow stuff unit vector	R_3	4	-0.89952
		5	0.42830
		6	0.07592
		7	-0.04080
Non-such unit vector	R_4	4	-0.01594
		5	0.13068
		6	-0.45187
		7	0.88232
Bad data thresholds	C_4		16
	C'_4		-8
	C_3		+8
	C'_3		-32
	C'_2		-24
	C_{12}		220
	C_{21}		0

TABLE 1. (Continued)

<u>Name</u>	<u>Symbol</u>	<u>Landsat 2 Band</u>	<u>Coefficient Value (s)</u>
Cloud Threshold	C_C		145
Water Thresholds	C_{W1}, C_{W2}		64, -10
Cloud Shadow Threshold	C_S		10
Atmosphere effect	α_1	4	1.2680
	α_2	5	1.0445
	α_3	6	0.9142
	α_4	7	0.7734
	X_1^*	4	60.9
	X_2^*	5	64.7
	X_3^*	6	75.7
	X_4^*	7	29.7
Yellow shift reference	Z_3^*		8.5

Data Average Measurement

From the set of good pixels calculate

$$\bar{Z} = \begin{pmatrix} \bar{Z}_1 \\ \bar{Z}_2 \\ \bar{Z}_3 \\ \bar{Z}_4 \end{pmatrix}$$

where the $\bar{Z}_i$ are simple averages over these pixels.

Soil Mean Measurement

From the good pixels calculate

$$\bar{Y} = \begin{pmatrix} \bar{Y}_1 \\ \bar{Y}_2 \\ \bar{Y}_3 \\ \bar{Y}_4 \end{pmatrix}$$

where

$$\bar{Y}_1 = \bar{Z}_1$$

$$\bar{Y}_2 = \text{lower 5\% level of histogram of } Z_2$$

$$\bar{Y}_3 = \text{lower 5\% level of histogram of } Z_3$$

$$\bar{Y}_4 = \bar{Z}_4$$

Green Arm Measurement

$$\bar{W} = \begin{pmatrix} \bar{W}_1 \\ \bar{W}_2 \\ \bar{W}_3 \\ \bar{W}_4 \end{pmatrix}$$

where

$$\bar{W}_1 = .73 \bar{W}_L - .68 \bar{W}_A$$

$$\bar{W}_2 = .68 \bar{W}_L + .73 \bar{W}_A$$

and where

$\bar{W}_A$ = upper 5% level of Z_A

$\bar{W}_L$ = average of Z_L values included above $\bar{W}_A$

$\bar{W}_3$ = average of Z_3 values included above $\bar{W}_A$

$\bar{W}_4$ = average of Z_4 values included above $\bar{W}_A$

2.1.3 Correction Procedures

Correction procedures consist of two steps, sun zenith angle correction and haze correction.

Sun zenith angle correction can be accomplished by utilizing the following relation:

$$x' = \frac{\cos \theta_0}{\cos \theta} x$$

where

θ_0 is a reference angle, and θ is the solar zenith angle for the given acquisition.

Haze correction, with the XSTAR algorithm uses the following relation:

$$x'' = Ax' + B$$

where A is a diagonal matrix and B is a vector.

The formula for the diagonal elements of A is

$$A_{ii} = e^{-a_i \gamma}$$

and for the elements of B it is

$$B_i = (1 - e^{-a_i \gamma}) x_i^*$$

where γ is chosen to minimize the following function.

$$\min \left\| Z_3^* - \sum_{i=1}^4 R_{3,i} e^{-a_i \gamma \bar{X}_i} + \left(1 - e^{-a_i \gamma} \right) X_i^* \right\|^2$$

where

$R_{3,i}$ are the components of the R_3 (yellow stuff) unit vector, and
 $\bar{X} = \overline{RW}$.[†]

In summary of the screening and diagnostic procedures, each pixel is corrected to the extent known a priori (that is, satellite corrected and cosine corrected but not haze corrected) and projected onto the Tasselled Cap coordinate system. In this coordinate system certain standard screening procedures are carried out. The pixels which are still labeled good after passing the screening tests are used to calculate diagnostic characteristics, primarily for the purpose of estimating the amount of haze present in the scene.

The output of screening is the original raw data, untransformed and unmodified in value, but with an additional channel which is used to flag the data status (cloud, water, etc.). The diagnostics are transformed back so as to be compatible with the original raw data and become part of the data description which is carried with the data as header information. The diagnostics are certain average points in the data structure. If later an improved satellite transformation is developed or an improved sun angle correction, these diagnostics will still be valid, since they are taken with respect to raw (untransformed) data.

After screening, the haze correction is calculated and then all of the corrections are applied to the data in one step to avoid developing round off errors in the 8-bit data words.

[†]The green arm feature was actually used in the corrections used in this work. However, the data average feature $\bar{Y}$, appears to be equally useful, is easier to calculate, and is preferred by P. Lambeck. The latest version of the XSTAR algorithm is documented in Reference 5. The above expression is approximated by a quadratic form for ease of calculation.

2.2 FEATURE EXTRACTION

The two main feature extraction steps are spectral and spatial. The spatial feature extraction step, in current practice at ERIM, usually is carried out on registered multitemporal data.

2.2.1 SPECTRAL FEATURE EXTRACTION

For the work reported here the spectral features that have been extracted are the first two Tasselled Cap features, brightness and green applied after haze correction. For passes which are late in the season for wheat (4th biowindow) we have also retained the 3rd Tasselled Cap feature, yellow stuff. (This is a minor point in as much as we have concentrated on acreage estimates during the first 3 biowindows.)

As was mentioned earlier, this step is combined with the data correction step above so as to minimize round-off errors.

2.2.2 SPATIAL FEATURE EXTRACTION

Landsat data over agricultural scenes has the characteristic that it tends to be homogeneous in fields. Each field may contain anywhere from a few to 100 or so pixels, so an opportunity exists to compress the Landsat data considerably, depending on the application. Furthermore, in the process of compressing the data an opportunity exists to average over the system noise.

How much can be gained in the way of system noise reduction depends primarily on how strong and recognizable the scene spatial structure is. If, for an extreme example, all fields were known to be 40 acres and arranged in a uniform checkerboard pattern, then this spatial structure could be exploited to average exactly over the pixels in each field. Very little of the total information in the scene would be "used up" by the process of fitting the data to the required $1/4$ mi x $1/4$ mi grid.

In the real world however, fields vary in size, and many are irregular in shape. Hence, the process of defining fields by finding their boundaries depends upon very local information, e.g., on the differences

(gradients) between neighboring pixels. Much of the total information in the scene gets used up by such a process.

When one examines an image of an agricultural scene in Landsat data one is impressed with the large number of straight boundaries which are evident. It would be interesting to attempt to explore these boundaries. Yet our opinion is that most of the useful information is contained simply in the concept of "nearby". Pixels which are near to each other are more likely to be the same class than pixels which are far away. Using this concept avoids the difficulty of explicitly defining boundaries.

Our approach in spatial feature extraction is to exploit this nearness idea to the extent that we can in a natural and economical way. We want to do this by grouping together pixels which are spectrally and spatially similar. Further, if fields are small, so that very few large groups exist, we would like our procedure to default to the processing of single pixels.

The procedure we have adopted to carry out these functions is, briefly, as follows.

1. To each pixel append additional channels which contain the spatial coordinates of the pixel, i.e., the line and point number of the pixel.
2. Cluster the pixels in any reasonable conventional way, using both spectral and spatial values in computing the distance measure.
3. Compute statistics for each cluster generated, including in particular the average spectral vector, the average spatial vector and the number of points included in the cluster. This information constitutes the compressed data.

The computer program which carries out these operations is called BLOB [3],[4] and the clusters produced by it, blobs.

For purposes of acquiring training data it is desirable to use only the pure part of blobs, that is, only the pixels which are not contaminated by the signal from adjacent blobs. For this purpose we have defined "stripped" blobs. Stripped blobs are those from which any pixels which were adjacent to any pixels of a neighboring blob have been deleted. The blob statistics are calculated only for the residue of undeleted pixels, and so are "pure" statistics.

All of the steps through spectral feature extraction, above, are carried out on individual passes of Landsat. Blobbing, on the other hand is carried out on registered multitemporal data. (Some further notes on blobbing will be found in Section 4.2 and Figures 7 and 8.)

PROCEDURE B TRAINING AND PROPORTION ESTIMATION

We have discussed the first major element in any remote sensing inventory system; namely, preprocessing/feature extraction. The next major question is, how are these features to be used to derive the required outputs of the inventory? In our case, the acreage proportions of wheat are estimated for each of the available 5 x 6 sample segments. Conventionally, one proceeds by obtaining labeled samples, using the samples to estimate the feature density function for each of the classes to be estimated, and then making a proportion estimate using that estimated feature density function and the rest of the available (unlabeled) samples. For example, one may classify the individual samples and then aggregate the classification to obtain a proportion estimate. In the simplest version of Procedure B all of these steps are combined in a 2-part procedure of selecting training and estimating proportions. The procedure can be viewed conceptually from a variety of perspectives as will be indicated later. First, however, we will simply describe the baseline procedure as it stands.

Table 2 outlines the two parts of the procedure, and these are described in more detail in the following sections. Referring to Table 2, the first step in training selection is to group together all of the blobs from the segments being processed, and perform an unsupervised clustering of the blobs, based upon the spectral components of the blob feature vectors. The result of this clustering is to perform a spectral stratification of the data. Every blob is assigned to some specific cluster. Next, a two-way table is made, segment by cluster. The number of blobs in a particular cluster in a particular segment is the entry in the table. Using this table, combinations of segments are chosen and evaluated. An attempt is made to find the combination of segments which will contain blobs in more spectral clusters than any other combination of segments.

TABLE 2. PROCEDURE B OUTLINE

Section

2. Image Preprocessing/Feature Definition
3. Procedure B Training Selection
 - Select segments and fields (blobs) within those segments
 - 3.1 Segment Selection
 - Cluster the blobs in all segments, creating spectral strata
 - Select segments so as to contain blobs which cover all the spectral strata
 - 3.2 Field Selection
 - Randomly select blobs from those which can potentially be used to cover all the spectral strata
4. Proportion Estimation
 - Identify the selected fields, estimate the proportion of wheat in each spectral stratum, and aggregate the total strata proportions
 - 4.1 Identify selected blobs
 - 4.2 Estimate the proportion of wheat in each spectral stratum
 - 4.3 Make a proportion estimate for each segment and for the entire group of segments.

Having selected training segments it is then necessary to select, as training samples, individual blobs within those segments. The total number of blobs to be selected is first spread out among each of the spectral strata (since it is desirable to have more training from larger spectral strata). The allocated number of training blobs for each spectral stratum are further allocated to the selected training segments. Within each training segment/spectral-stratum combination the training blobs are drawn randomly from those available.

3.1 SEGMENT SELECTION

Several auxiliary concepts will help in describing the segment selection routine.

Number Of Segments To Select

The number of training segments to be selected is a parameter of the procedure, say, M . M can range from one to the total number of segments. Typically we have been working with 4-6 training segments.

Pairwise Search Pattern

The optimal way to choose the set of M training segments is to evaluate all combinations of the K segments available for training, M at a time. The computer time required to accomplish this is unreasonably large. At the other extreme the segments could be selected stepwise one at a time, in a manner similar to a stepwise regression procedure. Thus, one first picks the best single segment. Then given that segment, one picks the next segment to give the highest value of the criterion function, etc. This procedure usually would give reasonable results at minimal cost. A compromise which is guaranteed to give better results but is still quite efficient is to pick the best pair of segments, then the best complementary pair and so on until M segments have been selected. The program actually developed allows the user to specify the number of segments to be selected at each sequential step.

Criterion Function

Initially we can think of an evaluation or criterion function which increments by one the first time that a spectral stratum becomes represented in the process of choosing segments. Thus, if there are 50 spectral strata, the maximum value the criterion function might achieve is 50. However, additional considerations have been built into the criterion function.

Lower Bound

If a segment only contained a single blob which represented a certain spectral stratum, it might be considered an accident. It might turn out on subsequent examination that the single blob was ambiguous in its definition or too small to be clearly defined. Thus, we might not want to claim the spectral stratum had been represented if only one (or indeed, only a few) blobs were chosen. Hence, the "lower bound" is a parameter of the selection program, and is under user control. We have generally operated the program with the lower bound equal to 2, which means that 2 blobs must be present in order to consider a spectral stratum to be represented.

Extra Weight

Even if a fairly large number of blobs is available to represent a particular spectral stratum from one training segment, one still might feel it is desirable to obtain some additional representation from another training segment. In particular, a spectral stratum might consist of both wheat and non-wheat, with wheat dominating some segments and non-wheat dominating others, and picking more than one segment would improve the chances of representing both cases. In order to accommodate this desire the criterion function allows the user to set weights on the first, second, third, etc., occurrence of representation of a spectral stratum by a selected segment. Most usually we have been assigning successive weights of 100, 20, 4, 0, to successive representations.

3.2 TRAINING FIELD (BLOB) SELECTION

In the process of selecting segments, knowledge of the number of blobs in each potential training segment/spectral stratum combination is required. The result of the segment selection is to choose a certain number of blobs for training from each selected training segment/spectral stratum combination. Within that combination the required number of training blobs are then chosen at random.

Randomization Of Blob Selection

The random selection order of blobs is actually introduced earlier, prior to the spectral clustering step. At that point all the blobs chosen to be clustered, from all the segments, are placed in a random order, so that no particular sequencing over segments becomes associated with the clusters which are produced. Prior to randomization the segment number is associated as a tag with each blob. After the spectral stratification is complete, the blobs must all be sorted by segment and spectral strata in order to select training segments. However, this sorting out is independent of blob number so that a random order is still maintained within each segment/spectral cluster combination. Therefore, blobs are merely chosen sequentially from this randomly ordered list.

Big Blobs vs. Little Blobs

Earlier, when we described the process of blob feature extraction, it was mentioned that, for training purposes, it is desirable to use only "pure" pixels. Pixels which have one or more neighbors belonging to a neighboring blob are therefore "stripped", i.e., labeled as a blob boundary pixel. Only those blobs which have one or more non-boundary pixels are used for training purposes, and the blob statistics (spectral mean vector) are computed only over the non-boundary pixels.

At this point it is well to mention that the spectral stratification is established by clustering only the big blobs. The small blobs are classified into these strata and are used in proportion estimation, but do not enter into the training selection.

PROPORTION ESTIMATION

As was mentioned earlier, there are a number of reasonable methods of estimating proportions once training fields have been established. Here we describe the one currently used with Procedure B. The basic approach is to identify the selected blobs, use these identifications to make a proportion estimate for each spectral stratum, and then extend these estimates over all of the segments.

4.1 IDENTIFY SELECTED BLOBS

The identifications to date have been based on ground truth maps provided for LACIE Phase II Blind Sites in Kansas. In an operational version of Procedure B, one would have analysts provide blob labels. However, in a research version there are two main reasons for using ground truth labels. First of all, it is desirable to separate errors created by the procedure under development from errors which might be introduced from mistaken labels. Secondly, it is desirable to have exhaustive labels (for all the blobs in the sites) available, for a reason which follows, and the only source of exhaustive labels is the blind site ground truth.

The reason for having exhaustive labels available is in order to examine a variety of parameter settings and to introduce random replicates of the procedure for testing and evaluation. In an operational environment the analyst would be called upon to identify a specific set of blobs for a single iteration of the entire procedure. Even though this is quite a bit of work, a large training gain could still be achieved relative to training and classifying every segment. But in the research environment it is less total effort to identify all of the blobs once and for all, prior to running the entire procedure many times.

An additional reason for exhaustive labeling is to be able to systematically locate any sources of error in the development of the procedure.

Techniques of Locating and Identifying Blobs from Ground Truth

Several approaches have been used in the process of establishing the identity of blobs from the ground truth. The current standard is described as follows.

Available Information

The ground truth identifications are made available to us in the form of a dark and light contact print (ammonia process) of the original ground truth aircraft photos. Normally these are acquired from an altitude such that 4 photos are needed to cover a 5x6-mile segment. Each photo is at a slightly different scale and a slightly different camera angle. Analysts at JSC have previously established and drawn the sample segment boundaries on an overlay of the photos and the fields labeled as wheat have also been outlined. Fields are labeled as wheat or as other major crops on the overlay.

The MSS data is processed by us into a variety of forms and presented on a line printer output at 8 lines per inch (producing an image at approximately 1:24000 scale). The variety of forms includes graymaps, blob maps and stripped blob maps.

The next step is to transfer the ground truth wheat field boundaries accurately from the separate photos onto an overlay to the line printer maps. Having done that, each big blob which intersects a wheat field is identified and the proportion of it which is wheat is estimated. If for any reason there is doubt as to the identity, a special symbol meaning "ambiguous" is used. These estimates are entered into a list which is stored and can be searched by computers. Any big blob not on the list is automatically considered to be non-wheat. Subsequently for any particular run of the segment and field selection procedures the identity of the selected fields is obtained automatically from the computer and the proportion estimates are made automatically.

The several types of line printer maps are used at various stages of the transferring and estimation process. The graymaps are used

primarily to find the positions of the mile roads in the image. Two overlays with a prepared mile road spacings are used* and adjusted to the best fit position by eye. The intersections of the mile roads are then recorded and used as reference points in the ground truth photos. Figure 6 shows a graymap of this type with the mile roads drawn on.

The particular graymap algorithm used for these maps is chosen to have a reasonable fidelity to the photo tones, but especially seems to heighten the contrast along the road network. The mapped quantity is,

$$m = \begin{cases} 201 - Y_{G,3} & , \text{ if } Y_G > 1 \\ 180 + Y_{B,3} & , \text{ otherwise} \end{cases}$$

where:

$Y_{G,3}$ is the green feature in the third biowindow (i.e., the 2nd component of $R^T X$ ", where X " is the corrected Landsat counts vector)

$Y_{B,3}$ is the brightness feature in the third biowindow.

The quantity m is mapped such that the lightest areas on the map correspond to the largest values of m . With this rule, green fields appear as dark to medium areas on the map. Fields which are not green enough to be interesting are mapped in respect to brightness, and appear from medium to light areas on the map.

Next the photo copies are mosaiced together (approximately, leaving an overlap at the boundaries) and the mile road grid is drawn onto each of the separate quarters of the mosaic. A reference intersection near the center of the 5x6-mile sample segment is clearly marked on both the photo copies and on the graymap overlay, so that any intersection can be easily located on both media.

*The basic technique of using mile road grid was developed by S. Lindner.

Two overlays are required since the angle between N-S roads and E-W roads as seen in Landsat data is a function of the latitude.

ORIGINAL PAGE IS
OF POOR QUALITY



FIGURE 6. GRAYMAP OF SEGMENT 1852, THIRD BIOWINDOW, COMPOSITE [BRIGHTNESS/
MINUS GREEN] RULE, DESIGNED TO ENHANCE MILE ROAD POSITIONS

From this point on several different procedures have been used, and we will describe two of these.

In the first procedure, the field boundaries and prominent scene features are transferred freehand to the graymap overlay. This is accomplished using the mile road grid as a guide. This procedure is relatively quick and is accurate to within a few pixels which is sufficient for areas of large fields. The overlay is then placed on top of the stripped blob map and those blobs which intersect a wheat field are labeled.

The blobs themselves are numbered in the process of producing them, and this number is coded into a symbol for the blob which is used in making the blob map. Two maps are used, one containing alphanumeric symbols corresponding to the numbers 1...50, and the other containing the symbols 0, $\emptyset$, 1, $\bar{1}$, ..., $\mathcal{9}$ corresponding to 0, 50, 100, 150, ..., 950. Together these cover the range 1 to 1000 and if there are blobs with numbers larger than 1000 they are easily identified by the portion of the map in which they fall. Figures 7 and 8 show a pair of blob maps of the types just mentioned.

The second general approach is to digitize the mile road intersections as control points and then digitize the vertices of all of the wheat fields on the photo copies. The digitized points can then be transformed to line printer coordinates through a computer program and an overlay can be drawn using an automatic plotter. The plot can be overlaid onto the blob maps as before and the blobs labeled. This method is, in principle at least, somewhat more accurate than the freehand sketching method, but it is also more time consuming. In either case the task of determining the identity of individual blobs is still the same.

One potential advantage of the digitizing method is that a computer program may be written to automatically determine the label for each blob from the digitized information. This would be especially valuable if different blob maps were to be made of the same region for testing purposes.



FIGURE 8. STRIPPED BLOB MAP, SYMBOLS FROM 0 through 19

In order to maintain perspective we note again that this discussion all applies to the research environment. The operational environment for the use of blob and Procedure B in general is quite different and in many ways less complicated. We will discuss the possible operational implications in Section 6.

4.2 ESTIMATE THE PROPORTION OF WHEAT IN EACH SPECTRAL STRATUM

The method of estimating the proportion of wheat in each spectral stratum is to divide the total number of wheat blobs among the sampled blobs by the total number of wheat plus other blobs. "Ambiguous" blobs are excluded from either numerator or denominator. (If a blob is 0.80 wheat then 0.8 is added to the numerator and 1.0 is added to the denominator.)

In symbols,

$$\lambda_i = \sum_{j \in \{N_i\}} b_{ij} / N_i$$

where

- λ_i is the estimated proportion wheat for the i^{th} spectral stratum
- b_{ij} is the estimated proportion of wheat in the j^{th} blob among those chosen for training the i^{th} spectral stratum
- N_i is the number of blobs in the training set $\{N_i\}$ for the i^{th} spectral stratum. Note that this set can extend over several segments.

4.3 ESTIMATE SEGMENT PROPORTION

The estimated proportion for each segment is calculated from the proportions of the spectral stratum of which it is comprised, as follows:

$$\hat{p}_s = \sum \lambda_i n_{is} / \sum n_{is}$$

where

$\hat{p}_s$ is the estimated proportion wheat in segment S

n_{is} is the number of pixels contained in blobs of segment S which belong to spectral stratum i.

We have now completed the description of the simplest version of Procedure B. In a later section we will discuss modifications and improvements to this basic system. In the next section, Section 5, we discuss results for the base line procedure carried out on 17 Blind Test Sites in Kansas.

RESULTS WITH BASELINE PROCEDURE B

An initial demonstration of Procedure B was carried out using manual procedures at all steps after the spectral strata clustering. In particular, the segment selection was carried out by lining up markers on the rows and columns of a large sheet of paper displaying the array of blobs contained in each spectral-stratum/segment combination. The required number of blobs for each combination were established roughly proportional to the square root of the total number of blobs in each spectral stratum, and roughly proportional to the number available for training in that stratum from each training segment. The required number of blobs were identified from a stripped blob map such as Figures 6 and 7, but without the assistance of a good greymap. The blob proportions, and the spectral stratum proportions were tabulated manually. The segment proportions were tabulated by a program which has since been superseded.

The initial manual operation was demonstrated on 17 sample segments, using six of them for training. The segments are those '75-'76 blind sites in Kansas for which acquisitions were available in or near the first three biowindows. The procedure as initially implemented did not incorporate ancillary data, either in a partitioning step or as a part of the classification methodology.

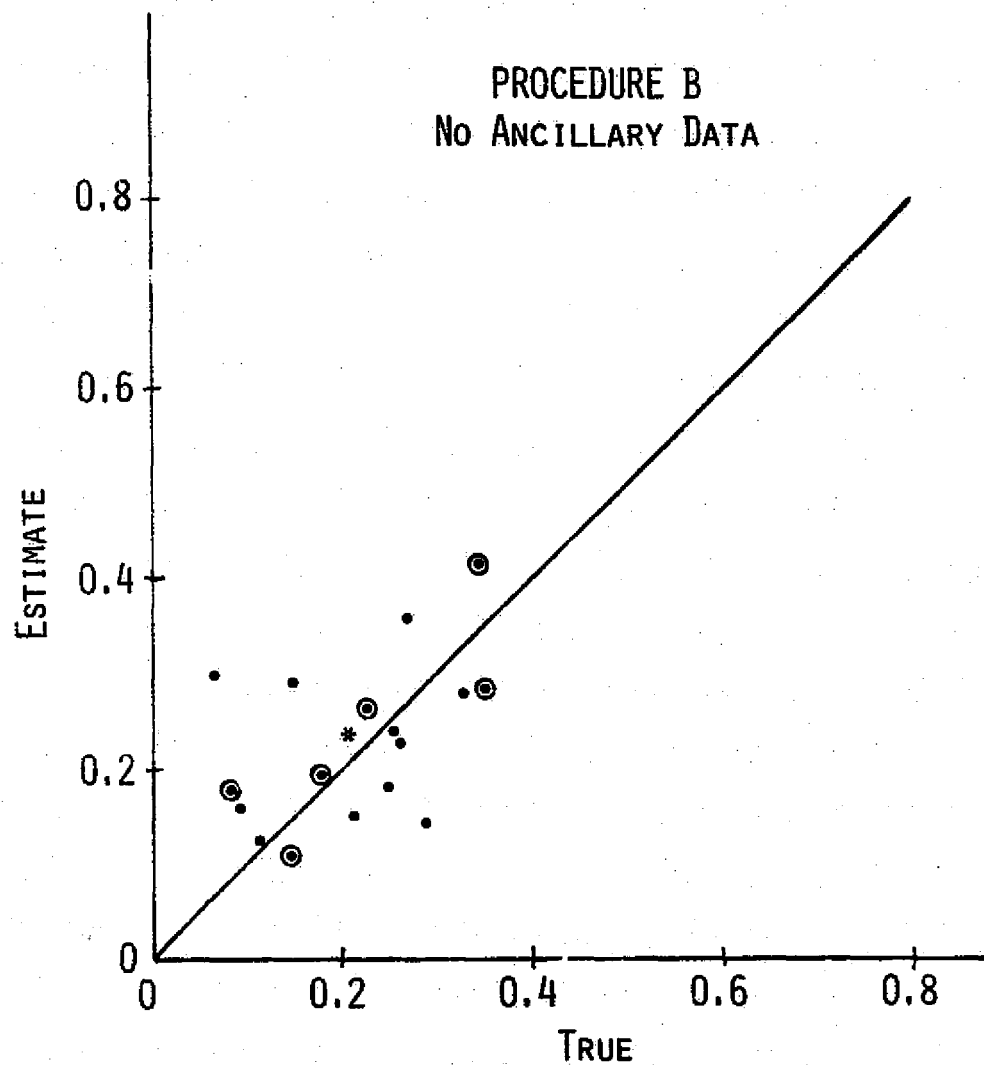
The performance of Procedure B on the 17 sample segments is summarized in Table 3, and in Figure 9. In the table the training segments are marked with an asterisk. The overall performance is indicated by the bias figure, 2.16%, and the coefficient of variation, which is the standard deviation of the error in percentage estimate, 9.12%, divided by the true percentage 20.93%, for a c.v. of 0.44.

The standard error of the overall bias estimate is $9.12/\sqrt{17} = 2.21$. Because the overall bias is less than this, it is not statistically significant.

TABLE 3. PERFORMANCE OF PROCEDURE B ON 17 SAMPLE SEGMENTS

Segment	Estimated Percentage, E	True Percentage, T	E - T	
1020	23.0	25.29	- 2.29	
*1035	19.65	17.53	2.12	
*1041	11.1	14.39	- 3.29	
*1154	28.8	34.8	- 6.00	
1163	16.0	8.72	7.28	
1165	29.95	6.51	23.44	
*1167	17.7	8.03	9.67	C. V. = 0.44
1180	36.0	26.36	9.64	
1851	24.5	25.0	- 1.00	
*1852	26.7	22.29	4.41	
1860	18.6	24.8	- 6.20	
*1861	41.9	34.39	7.51	
1865	15.2	20.4	- 5.20	
1883	28.2	14.4	13.80	
1886	14.55	28.85	-14.30	
1887	12.8	10.9	1.90	
1891	28.4	33.18	- 4.78	
Average	23.12	20.93	2.16	
σ	8.54	9.32	9.12	

If we have trained sufficiently, the performance of the six training sites ought to be indistinguishable from the performance of the 11 recognition sites. Table 4 shows summary performance for these cases. The performance for the training segments has slightly higher bias and somewhat smaller standard deviation.



KEY: ● TRAINING SEGMENT
* MEAN OF ESTIMATE VERSUS MEAN OF TRUE

FIGURE 9. ESTIMATED PROPORTION VS. TRUE PROPORTION FOR
17 BLIND SITES IN KANSAS

TABLE 4. PERFORMANCE SUMMARY FOR TRAINING AND RECOGNITION
SEGMENTS CONSIDERED SEPARATELY

	<u>6 Training Sites</u>	<u>11 Recognition Sites</u>
Average True Percentage	21.91	20.40
Average Estimated Percentage	24.31	22.43
Bias	2.40	2.03
Standard Deviation	6.10	10.69
Standard Deviation of the Mean	2.49	3.22
Coefficient of Variation	0.28	0.52

In both cases the bias is within the error of estimate of the mean that one would expect from these few samples, so there is no significant bias for the two cases. Neither is there a significant difference between the biases.

With regard to the standard deviations, an F test of the ratio of the two variances shows that they are not significantly different at the 5% level.

The coefficient of correlation between the estimated and true proportions for all sites is 0.48, which is significantly different from 0 at the 5% level.

These results can be interpreted as a moderate success for Procedure B, one which suggests that variations on the procedure ought to be systematically explored. Therefore, many aspects of the procedure have been automated and many runs have been made with varied parameters. So far none of those runs has duplicated the success obtained on the manual exercise.

Figures 10 and 11 are examples obtained for two different cases. Figure 10 is a three-pass case with 17 sample segments and five training segments. Figure 11 consists of ten carefully chosen segments (i.e., closely matching acquisition dates) and three training segments.

At first it was thought that a particular training segment which was missing in Figure 10, compared to Figure 9, was the cause of the trouble. However, it has been found that the results do not appear to be particularly sensitive to the choice of individual segments. Subsequently we have checked many aspects of the automated procedure and the identification of ground truth. Errors have been found but none which constitute an explanation.

A remarkable consistency of pattern is present among all the results. Certain segments whose ground truth proportion is low have very high estimates. Certain segments whose ground truth proportion is high have very low estimates. The same segments display the same bias over a variety of parameter settings. Thus Figure 11 looks like a random sample drawn from Figure 10.

If we could throw out these few bad points the overall correlation looks quite reasonable, but there is no rational basis for throwing out samples in the absence of ground truth. [Appendix IV, added in proof, gives an explanation for these bad points and shows how they can be brought into line by the use of an ancillary variable.]

There remain two differences, which have not yet been checked, between the manual exercise and the subsequent automatic ones. The manual exercise was conducted on blobs generated using four acquisitions, whereas the automatic exercise has been conducted on three-acquisition blobs. Secondly, there were many potential sources for error in identification of the ground truth during the manual exercise, and so only blobs which were obviously identifiable were used for training. In many

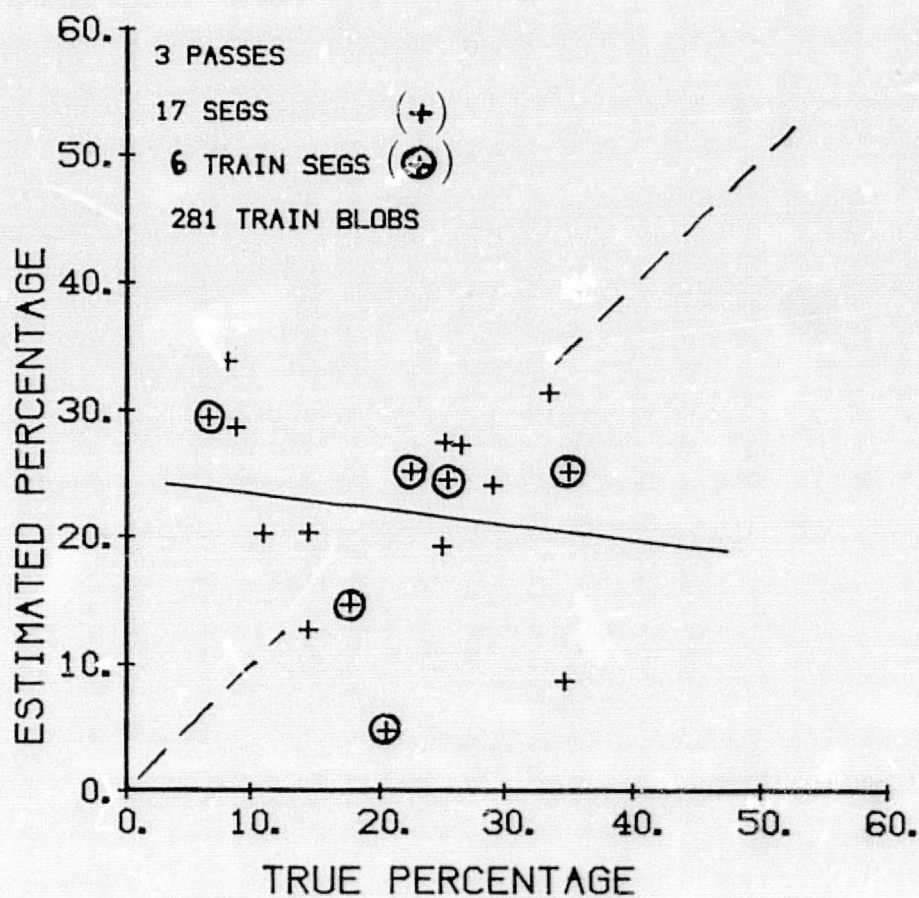


FIGURE 10. ESTIMATES VS. TRUE FOR 17 SEGMENTS AND 6 TRAINING SEGMENTS,
69 SPECTRAL STRATA

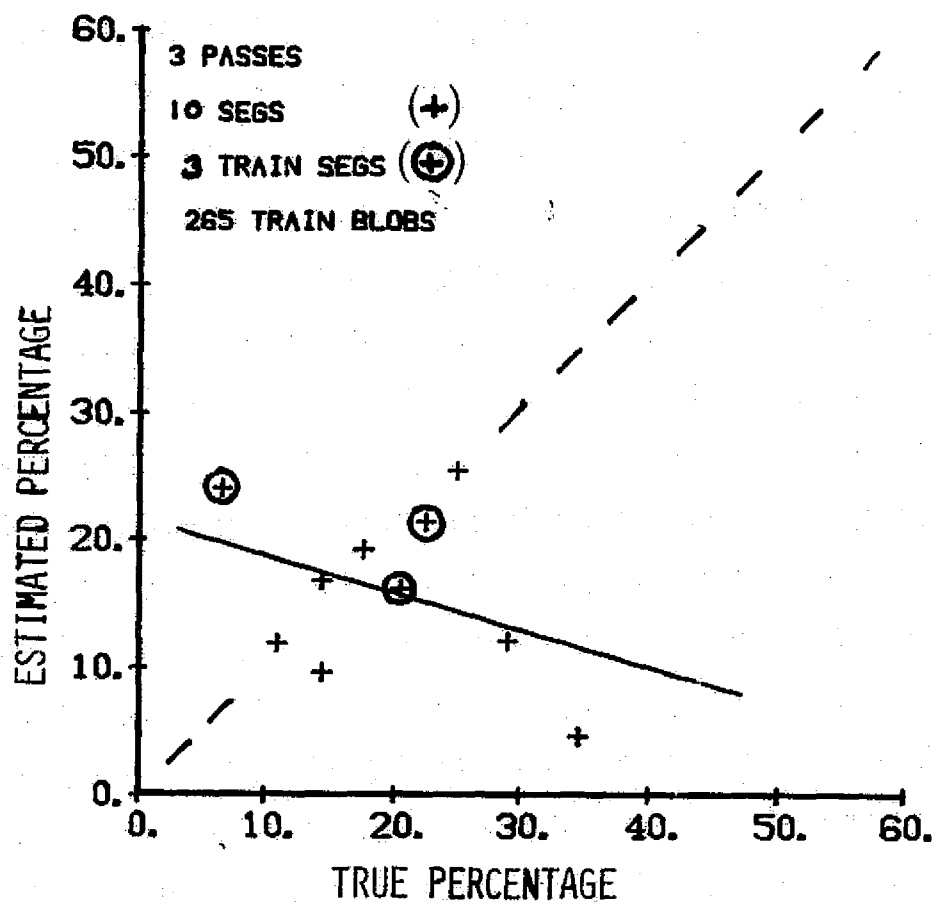


FIGURE 11. ESTIMATES VS. TRUE FOR A SUBSET OF 10 SITES CHOSEN TO HAVE CLOSELY MATCHING BIOWINDOWS, 47 SPECTRAL STRATA

cases this meant that only blobs with a fairly large number of pixels, even after stripping, were used. At the present time these last two sources of explanation are being studied.

EXTENSION TO THE BASELINE PROCEDURE B

In the foregoing we have described the simplest version of Procedure B as of the date of this report. Two major topic areas have occupied our interest in the way of extensions to this basic procedure, namely ancillary data dependence and operational system capabilities.

6.1 ANCILLARY DATA DEPENDENCE OF SIGNATURES

We outlined, in Section 1, the sources of variability in crop signatures. These included: external physical effects which can be accounted for if the parameters driving them can be measured (as in the cases of the satellite type, the sun zenith angle and the amount of haze), also included are random sources of variation which can be averaged over (as in the case of the training selection portions of Procedure B); and finally, systematic effects which are dependent upon ancillary variables which are measureable, but the magnitude and form of the dependencies are unknown initially. It is these last effects which we wish to discuss at this point.

Candidate Ancillary Variables

Ancillary variables fall into several main categories, as shown in Table 5.

The site specific ancillary variables describe permanent features of a segment. Pass specific variables describe conditions which can vary from pass to pass and among these we distinguish those that can be measured from the satellite image data itself, and those that can be measured from other sources such as meteorological satellites or an ephemeris.

Conceptual Model For The Use Of Ancillary Data

Two general ways have been proposed for handling the dependency of signatures upon ancillary data, partitioning and signature modeling. First, a few words about partitioning.

TABLE 5. GROUPING OF ANCILLARY VARIABLES

SITE SPECIFIC VARIABLES

- Average growing season temperature (degree days)
- Average growing season rainfall (inches)
- Latitude/longitude/ground elevation
- Land use category
- Soil classification

PASS SPECIFIC VARIABLES (EXTERNAL)

- Sun zenith angle/scanner viewing angle
- Crop calendar
- Accumulated degree days
- Accumulated rainfall
- Satellite

PASS SPECIFIC VARIABLES (INTERNAL, DIAGNOSTIC)

- Soil mean
- Green arm mean
- Haze diagnostic
- Percentiles of green development

The general idea of partitioning is that one finds an area or domain of the partitioning variables over which signatures are essentially constant. Within this domain training and proportion estimation takes place.

An assumption is that such clear strata exist. There are cases where strata are divided by obvious boundary lines, but more generally there is a smooth continuum of partitioning variables, and the actual position of partition boundaries is somewhat arbitrary.

Secondly, in order to do an adequate job of partitioning one would like to include all of the possibly significant ancillary variables into the definition of a partitioning rule. As the number of variables is increased the size of partitions tends to grow smaller and smaller, even though there also is much correlation among the partitioning variables, either in their occurrences, in their effects, or in both. The tendency of partitions to get smaller can be overcome by taking account of these correlations, but this in effect requires a signature model.

In Appendix III we develop a viewpoint which encompasses both ancillary variable dependence and partitioning. Partitioning is explained in terms of signature models, and this is intended to put a rational framework behind partitioning.

We think of the ancillary data as conditioning the likelihood functions for the various crops which may be present in the scene. Then we imagine that classification or proportion estimation is carried out in a conventional way using these conditional likelihoods. A partition is a special case in which the values of some of the ancillary variables are bounded and an average is taken over the bounded range.

Results With Ancillary Variables

We have attempted to incorporate ancillary variable dependence into Procedure B, but have not achieved acceptable performance to date.

We believe the primary reason has been that we have not taken proper account of the dimensionality of the ancillary data space, whether this is expressed by saying that we haven't yet selected the "correct" ancillary variable function, or by saying that the ancillary variables are highly correlated in their effects. Appendix III.3 examines this question of dependency in a mathematical sense. The important conclusion is that the intrinsic dimensionality of the ancillary data space is never more than the number of spectral signature parameters

which must be estimated. However, if this is simply the mean brightness and green for three passes for a single crop (i.e., wheat) that is 6 degrees of freedom. Thus, a minimal number of sites (i.e., different ancillary variable conditions) which might be used to estimate these 6 parameters is about twice as many or 12 of them.

The ancillary variable problem, the problem of signature dependence and the problem of signature extension are all fundamentally the same problem.

We have analyzed the signature extension problem as having three fundamental components, (external effects, random effects and ancillary variable dependent effects) each of which can be attacked by a separate and distinct methodology. Over the next two or three years the relative importance of these various components will become quantified. The relative balance will be different for different survey problems. It will probably be found that winter wheat is one of the easiest problems in regard to ancillary variable dependence and that in order to attack more difficult problems a systematic crop signature model will be a necessity. This in turn is going to require:

- a. an ongoing, centrally organized, comprehensive, well controlled field measurements program to obtain information on the dependence of crop signature on ancillary variables
- b. a comprehensive modeling effort to relate the crop signatures and ancillary variables
- c. a large scale ground truth effort to obtain empirical signatures as a function of a wide diversity of random and systematic effects
- d. an empirical operational model or experience organizer, i.e., a systematic methodology for converting ground truth and analyst interpretations into a computer signature model.

6.2. OPERATIONAL SYSTEM CAPABILITIES

In order for any system to become operational there are certain capabilities it must incorporate. Perhaps the most important is the ability to make timely estimates in the presence of missing or partial data. A desirable, although non-essential capability is to choose training efficiently in a time sequential manner. As of the writing of this report both of these capabilities are in a developmental stage. The next two subsections explain these developments in more detail.

6.2.1. THE PROBLEM OF MISSING ACQUISITIONS

Suppose data has been acquired for the first three biowindows for some segments; for other segments, however, data has been acquired only for various combinations of two of the three biowindows, while for still others data is available for only one of the three biowindows.

In its simplest form Procedure B could be applied to each of the sets of segments separately. However we would expect a greater net training gain and less potential for confusion in an operational system if Procedure B could be extended to operate on all the segments even with missing acquisitions. A method of accomplishing this has been devised and is described as follows.

- a. During the process of clustering of blobs to produce spectral strata the blobs which have a full complement of acquisitions are clustered first. After the full clusters have been established the clustering continues using the blobs from segments with missing acquisitions.
- b. A blob containing partial acquisitions is assigned to a full acquisition spectral stratum based just on the available spectral coordinates. For example:

If three passes are available;

Three-pass mean,

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix}$$

Three-pass covariance,

$$\begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix} = \Sigma$$

Distance measure is

$$(X - \mu)^T \Sigma^{-1} (X - \mu)$$

If only passes 1 and 2 are available, ignore the third component of the pixel vector and strike out the third row and third column of Σ

- c. If the partial acquisition blob is not near enough to any of the existing spectral clusters, a new cluster is defined involving only the partial acquisitions.
- d. After the clustering is complete the segment selection procedure is exercised as before, the main difference now being that full complement blobs are counted as belonging both to the full complement spectral strata, plus all subsets of them. (see Table 6) Thus, segments which have a full set of acquisitions are represented in a large number of strata and are thus more likely to be chosen for training.
- e. With these modifications the estimation procedure is carried out just as before.

TABLE 6. EXAMPLE REPRESENTATION OF SUB-STRATA. THE ENTRIES IN THE TABLE ARE THE NUMBER OF BLOBS REPRESENTED BY THE SEGMENT/SPECTRAL STRATA COMBINATION

	<u>Segment</u>		
	<u>1</u>	<u>2</u>	<u>3</u>
1, 2, 3	5	0	0
1, 2, -	5	3	0
1, -, 3	5	0	0
-, 2, 3	5	0	0
1, -, -	5	3	4
-, 2, -	5	3	0
-, -, 3	5	0	0

6.2.2 THE SEQUENTIAL SELECTION PROBLEM

In an operational system it will be desirable for various reasons to use the same segments for training on successive acquisitions, to the extent possible. It may be that the identifications of training fields from an earlier biowindow may still be valid at a later time and so analyst effort can be saved. There may be some benefit from the analyst developing familiarity with a segment so that the segment is easier for him to work the second time around. The segment selection program could be modified to include weights on the individual segments so that, other things being equal in the choice between two segments, the one which had previously been selected for training would be selected again.

A further possibility exists. Could not the program be modified so that, other things being equal, the segment would be chosen which had the highest probability of being available on a subsequent (i.e., future) acquisition? In order for this to be a worthwhile capability to implement, there must be a means of estimating the probability of future

acquisition. One means depends on whether the segment is in the overlap zone between successive day passes of the satellite. If it is, the pair probability of at least one acquisition is nearly double the single day probability. This factor could be further modified by the position of the window relative to the 18-day overpass cycle of the satellite. The second factor is the quality of the multitemporal registration reference pass. This affects the probability of being able to register the data multitemporally for a given degree of cloud cover. These factors are subject to determination or estimate, so that a table of acquisition probabilities for each segment for each biowindow can be produced and updated each time the Procedure B segment selection program is to be run. (Note that the entries into the probability table are either 0 or 1 for past events, i.e., the acquisition either failed or was successful.) The future event entries in this table can vary by as much as a factor of two for a short biowindow.

Now the value of selecting a segment for training in the current acquisition is related to:

- a. whether it was acquired or not in a previous biowindow or combinations of them.
- b. whether it was used for training previously.
- c. what the probability of acquisition is for future biowindows or combinations of them.

In order to simplify the problem somewhat we will assume that the segments have a training value which is proportional to the number of acquisition combinations covered by the segment, namely $2^n - 1$, where n is the number of acquisitions. Thus, the expected value of a segment is equal to the probability of being acquired n times (by end of season) times $2^n - 1$, summed over the values of n . And the probability of being acquired n times is a sum over the different ways of acquiring it. Thus, for a case where a maximum of three biowindows are being considered the expected value of a segment is

$$\begin{aligned}
 EV &= (2^3 - 1)(p_1 p_2 p_3) \\
 &+ (2^2 - 1)(p_1 p_2 [1 - p_3] + p_1 [1 - p_2] p_3 + [1 - p_1] p_2 p_3) \\
 &+ (2^1 - 1)(p_1 [1 - p_2][1 - p_3] + [1 - p_1] p_2 [1 - p_3] \\
 &\quad + [1 - p_1][1 - p_2] p_3)
 \end{aligned}$$

where p_1 , p_2 , and p_3 are the entries in the probability table. If for example, this is the second acquisition we might have $p_1 = 0$, $p_2 = 1$ and $p_3 = 0.25$ giving

$$\begin{aligned}
 EV &= 0 + 0.75 + 0.75 \\
 &= 1.5
 \end{aligned}$$

Given two segments having equal actual training value as determined by the currently available acquisitions we will want to decide between them based on their expected value. Since we don't know in advance the trade off between training efficiency and EV we will have to define a relative weight between these factors and adjust it empirically. A convenient form is

$$\text{Total value} = (TV) * (1 + k * (EV))$$

where

TV is training value

EV is expected value

k is a relative weight factor

This function is being incorporated into a revised version of the segment selection program.

CONCLUSIONS AND RECOMMENDATIONS

Thus far the results achieved, i.e., proportion estimation performance, using Procedure B are inconclusive. However, in the immediate future we will continue to search out and correct possible sources of error in the procedure.

Certain statements ought to be made irrespective of the outcome of that evaluation.

The preprocessing/feature extraction structure we have set up appears to be stabilized. The structure is transportable in the sense that the ideas are simple and can, and have been, implemented by other researchers. There will, no doubt, be further improvements in the correction algorithms and these will be incorporated as they become available. But we believe that the present implementation forms a solid working structure.

We have characterized the signature extension problem in terms of three main sources of variation in signatures:

1. systematic external effects applicable to all crops
2. systematic ancillary variable effects on individual crops
3. apparently random variation between segments which appear identical in terms of the ancillary parameters being observed.

The value of this taxonomy of the problem is that three distinct methodologies are available to attack the three aspects of the problem, i.e.,

1. physical models combined with regression using unlabeled samples over a variety of conditions of measurement
2. signature studies on individual crops, using both canopy models, field measurements and empirical regression against the ancillary variables
3. multisegment training and statistical analysis of sampling strategies for training sufficiency and efficiency.

Evaluating progress in these areas in the SR&T community, we can say that much progress has been made in the first area. Specific improvements that are still needed and possible are discussed in [5]. In the second area, a comprehensive and sustained approach to the ancillary variable dependent signature problem is needed, as outlined in Section 6.1.

In the third area, we regard Procedure B as a prototype framework in which questions of a statistical nature concerning signatures, signature variability and sampling can be asked, and, we believe, answered. Procedure B appears to be soundly based, statistically, as a stratified sampling scheme. It shares with other signature extension methods a tendency for estimates to correlate well with each other, but less well with the ground truth. Thus a study of the sources of error in Procedure B will not only be expected to make the procedure work acceptably, but also to strike at the heart of the signature extension problem.

If there are no significant errors in the application of the procedure, then what we have found is a bona fide example of attempting signature extension over too large a partition (i.e., the entire state of Kansas). This would force the use of smaller partitions or the incorporation of ancillary data into the signature model in order to achieve acceptable performance. Appendix IV, added in proof, explores this possibility and presents some further results.

APPENDIX I

A TRANSFORMATION TO MAKE LANDSAT 1 MSS DATA COMPATIBLE
TO LANDSAT 2 MSS DATA

(R. Kauth, J. Hemdal)

1. INTRODUCTION AND DATA BASE

It was desired to estimate coefficients of a transformation which would convert Landsat 1 data to Landsat 2 data. In order to make the estimate it is desirable to use the identical scene observed under identical conditions by both satellites. The nearest we can come in practice is to observe pairs of data over the same scenes separated by nine days. An initial selection of about 25 such pairs was made, but natural attrition reduced this ultimately to eight pairs. As will be seen this is a quite minimal set and the fitting procedure should be repeated with a larger sample.

Pairs were eliminated from consideration if either member contained "patchiness" of cloud or haze, as evidenced in the histogram output of program SCREEN, or if the haze levels of the pair were thought to be markedly different from each other as evidenced by the yellow level of the mean of soils calculated by program SCREEN. The relative yellow level for each pass was judged against all non "patchy" passes for the same satellite by plotting a separate yellow level histogram over those passes for each satellite. The finally accepted pass pairs are listed in Table 1. Of the eight cases, four are cases in which the Landsat 2 pass precede the Landsat 1 pass.

The data used for fitting was the four band "mean of soils" and the four band "mean of the green arm", both outputs of program SCREEN. These data and their average are shown separately for each Landsat band in Table 2.

2. MODELS

Several different models were used for fitting. In general the models are of the form

TABLE 1. PASS PAIRS USED IN LANDSAT 1 : LANDSAT 2 FITTING PROCEDURE

<u>Case #</u>	<u>Segment #</u>	<u>L1 Date</u>	<u>L2 Date</u>	<u>Sun Zenith Angle, θ_1</u>	<u>Sun Zenith Angle, θ_2</u>
1	1020	155	164	37°	32°
2	1020	101	92	47°	46°
3	1154	153	162	37°	31°
4	1855	82	73	52°	52°
5	1855	82*	91	52°	46°
6	1857	101	92	46°	44°
7	1861	101	92	46°	45°
8	1882	153	162	37°	31°

* Same pass used twice.

TABLE 2. DIAGNOSTIC DATA USED IN FITTING

Band	Case	Soil Mean, YT		Green Arm Mean, WT		Average	
		L1	L2	L1	L2	L1	L2
4	1	41.5	41.7	25.1	24.4	33.3	33.05
	2	40.2	36.8	28.0	22.0	34.1	29.4
	3	36.1	36.3	26.0	23.6	31.05	29.95
	4	34.2	30.3	26.9	22.1	30.55	26.2
	5	34.2	36.3	26.9	26.3	30.55	31.3
	6	39.7	37.6	28.7	25.0	34.2	31.3
	7	43.6	41.2	35.0	32.0	39.3	36.6
	8	34.9	35.1	25.6	24.6	30.25	29.85
5	1	40.3	47.1	16.9	25.4	30.1	36.25
	2	39.6	41.7	24.3	23.3	31.95	32.50
	3	31.3	38.0	16.6	20.3	23.95	29.15
	4	33.5	34.3	25.4	25.8	29.45	30.05
	5	33.5	41.9	25.4	29.7	29.45	35.80
	6	40.7	44.8	25.2	28.8	32.95	36.80
	7	46.3	49.7	36.8	40.7	41.55	45.20
	8	29.2	36.4	16.2	20.6	22.70	28.50

TABLE 2. DIAGNOSTIC DATA USED IN FITTING (Continued)

Band	Case	Soil Mean, YT		Green Arm Mean, WT		Average	
		L1	L2	L1	L2	L1	L2
6	1	40.5	49.5	4.15	50.5	33.30	50.0
	2	38.8	41.7	40.7	40.5	39.75	41.1
	3	32.8	40.5	46.3	54.6	39.55	47.55
	4	32.3	34.3	32.5	33.9	32.4	34.1
	5	32.3	42.5	32.5	45.8	32.4	44.15
	6	38.8	45.1	44.3	46.1	41.55	45.6
	7	44.5	50.0	40.6	44.3	42.55	47.15
	8	34.3	41.0	43.5	59.6	38.9	50.30
7	1	17.1	20.5	22.0	25.3	19.55	22.90
	2	17.0	17.5	21.6	20.1	19.30	18.80
	3	14.0	16.1	25.8	26.7	19.9	21.40
	4	14.5	15.2	16.5	16.6	15.5	15.9
	5	14.5	18.0	16.5	22.3	15.5	20.15
	6	16.9	18.9	24.2	22.7	20.55	20.80
	7	19.8	20.9	19.1	19.3	19.45	20.1
	8	14.8	16.6	23.1	28.6	18.95	22.6

$$x_2 = F(\alpha, \theta_1, \theta_2, x_1) + \Sigma$$

where

x_2 is the Landsat 2 signal

x_1 is the Landsat 1 signal

θ_1 and θ_2 are the sun zenith angle of Landsat 1 and Landsat 2 cases, respectively,

α are a set of parameters which must be estimated

Σ is an error

Three specific models were tried, as follows:

a. Ratio model assumes for each band,

$$x_2 = A \frac{\cos \theta_2}{\cos \theta_1} x_1 + \Sigma$$

i.e., that there is no difference in signal offset between the two satellites, and that the radiance returned from the scene is an inverse function of the cosine of the sun zenith angle.

b. Offset model, assumes

$$\left(\frac{x_2}{\cos \theta_2} \right) = A \left(\frac{x_1}{\cos \theta_1} \right) + B$$

c. Three parameter offset model, assumes

$$x_2 = A_2 A_1^{-1} \frac{\cos \theta_2}{\cos \theta_1} x_1 + B_2 - A_2 A_1^{-1} \frac{\cos \theta_2}{\cos \theta_1} B_1$$

where

A_1 is the responsivity of Landsat 1

A_2 is the responsivity of Landsat 2

B_1 is the offset for Landsat 1

B_2 is the offset for Landsat 2

thus

$$x_2 = A \frac{\cos \theta_2}{\cos \theta_1} x_1 + B_2 - A \frac{\cos \theta_2}{\cos \theta_1} B_1$$

Model (a.) requires a fit to one parameter per band, (b.) requires two parameters per band, and (c.) requires three parameters per band. The residual error per band after fitting is shown in Table 3. The Model (c.) is considerably the best fit for band 7 and slightly better for the other bands.

TABLE 3. RMS ERROR OF THREE MODELS

<u>Band</u>	<u>Model a.</u>	<u>Model b.</u>	<u>Model c.</u>
4	1.11	1.05	0.81
5	2.06	1.60	1.36
6	2.99	2.24	1.66
7	4.78	3.39	1.20

In general we can write

$$z = a_0 + a_1x + a_2y$$

and identify

$$a_0 = B_2$$

$$z = x_2$$

$$a_1 = A, \quad x = \frac{\cos \theta_2}{\cos \theta_1} x_1$$

$$a_2 = -AB_1, \quad y = \frac{\cos \theta_2}{\cos \theta_1} x_1$$

In order to minimize the noise of the observations of the soil and green arm points we used their averages from Table 2. Thus, x_1 is the average L1 in Table 2. Using these tabulated values, and regressing z upon x and y , gives the results shown in Table 4.

Most times we will be interested in converting Landsat 1 data to appear as though it were Landsat 2 data at the same solar zenith angle. Therefore the model will simplify to the form

$$x_2 = Ax_1 + B$$

$$\text{where } B = B_2 - AB_1$$

B is also given in Table 4.

TABLE 4. REGRESSION COEFFICIENTS FOR THREE PARAMETER MODEL C

<u>Band</u>	<u>$a_0 = B_2$</u>	<u>$a_1 = A$</u>	<u>$a_2 = -AB_1$</u>	<u>B_1</u>	<u>B</u>
4	-19.48	1.04	13.69	-13.16	-5.79
5	-24.99	1.00	26.18	-26.18	1.19
6	-74.70	1.09	71.79	-65.86	-2.91
7	-21.53	0.82	24.54	-29.93	3.01

APPENDIX II

PROCEDURE B DATA FLOW

Figure 1 is a schematic of Procedure B data flow and programs. Not all of the listed programs have been described in the main text in this report since portions of the procedure are still under development. Following is a brief description of the purpose of each program.

<u>CONVERT:</u>	Transforms data from universal to ERIM format.
<u>SCREEN:</u>	Flags pixels for clouds/shadow, water, bad data; and computes diagnostics.
<u>MERGE:</u>	Combines data from separate passes and merges flags.
<u>CORRECT:</u>	Carries out pixel by pixel correction for satellite, sun zenith angle and haze, and tasselled cap feature extraction.
<u>BLOB AND STRIP:</u>	Carries out multitemporal-spatial feature extraction. Outputs a blob feature tape and a pixel tape.
<u>ACOV:</u>	Computes a metric for spectral clustering program (BCLUST).
<u>ROSSTP:</u>	Sorts blobs into big and small categories. Prepares a big blob file to be randomized.
<u>I7RAND:</u>	Randomizes order of blobs over an entire partition or stated collection of segments.
<u>BCLUST:</u>	Clusters blobs, producing a spectral stratification of them extending across segments.
<u>BLIST:</u>	Produces working files for following programs.
<u>WCT:</u>	Segment and blob training selection program.
<u>PRORP:</u>	Calculates estimated proportion for each spectral stratum.
<u>PROP:</u>	Calculates estimate proportion for each segment.
<u>GRAF:</u>	Produces graphs. (Courtesy of The University of Michigan's Aeronautical Engineering Department)

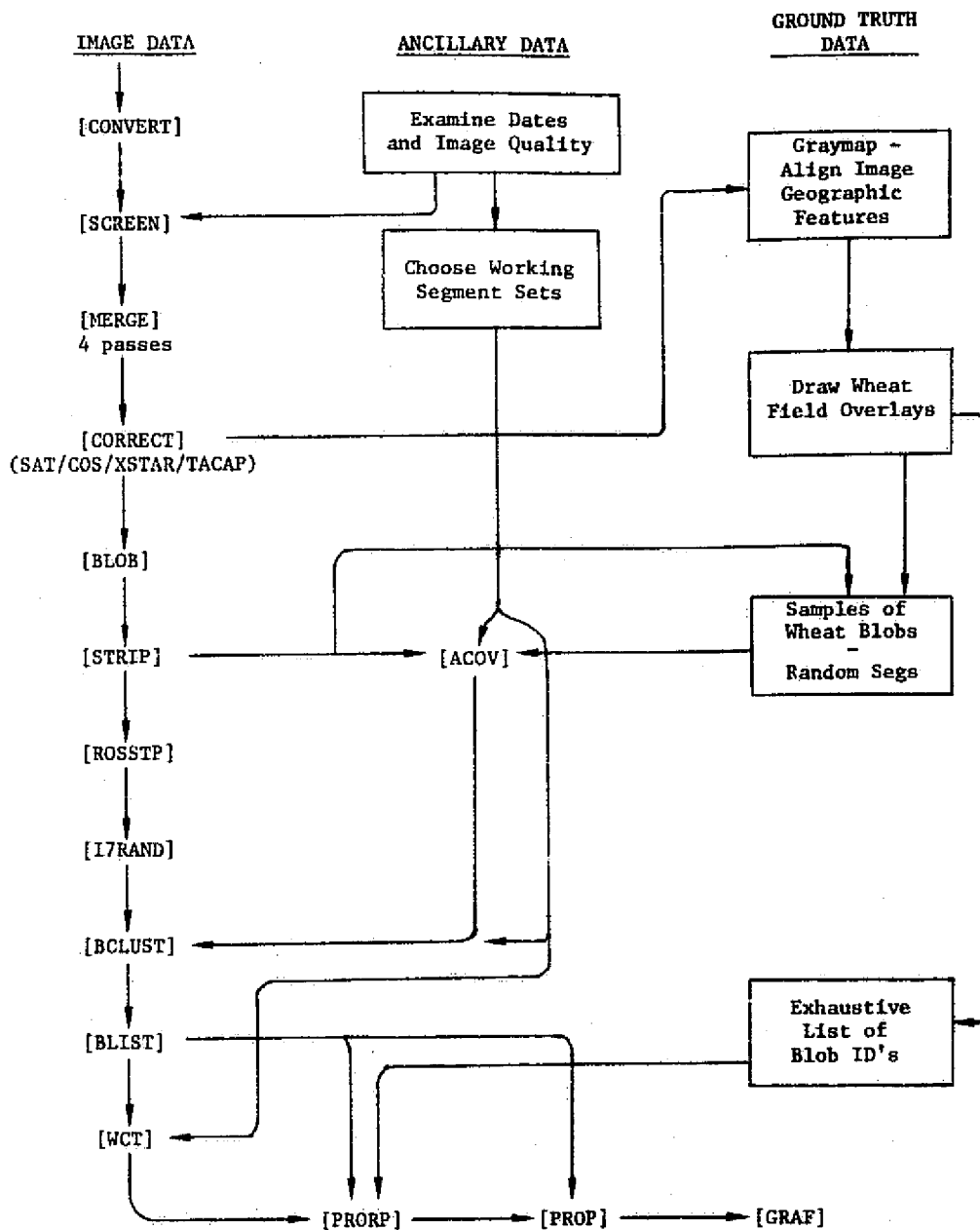


FIGURE 1. DATA FLOW AND PROGRAMS COMPRISING PROCEDURE B.
Program names are enclosed in square brackets.



Of these programs most are not of detailed interest to anyone outside of ERIM. The current versions of the screening and correction programs are documented in reference 5. Blob and BCLUST are documented from a users point of view in this appendix. The segment selection program, WCT, is documented in this appendix by means of an annotated Fortran listing. The linking and utility programs, PRORP, PROP and AERO:GRAF are actively being modified and would be quite dependent upon the particular computing installation, so they are not documented here.

APPENDIX II.1
USER DOCUMENTATION OF BLOB
(W. Richardson)

ROUTINE: BLOB

VERSION: 3.0

DATE: 25 March 1977

PROGRAMMER: W. Richardson W.R.

COMPUTER: 7094

LANGUAGE: MAD

INTERFACE: A POINT Module

PURPOSE: To group pixels into "blobs", that is, clusters that are spectrally homogeneous and spatially contiguous or nearly so.

COMMENTS: BLOB 3.0 is a rewrite of BLOB 2.0, which was programmed by A. Pentland and was based on an algorithm devised by R. Kauth with the assistance of G. Thomas and A. Pentland [3]. BLOB 3.0 runs 7 times as fast as the previous version.

DESCRIPTION:

The basic idea of BLOB 3.0 is the same as that of the previous version, namely, that the data is clustered (by an algorithm similar to A. Pentland's CLUSTER) using spectral channels and two additional channels containing the line and point number. Each pixel is processed as follows:

1. Find the current blob, say Blob j , that the pixel is closest to.
2. If the distance from the pixel to Blob j is less than a parameter τ , include the pixel in Blob j and update the mean of Blob j .
3. But if the distance is greater than or equal to τ , start a new blob containing one point (the pixel being processed) whose mean is the channel values of the pixel and the line and point number.

4. The line and point coordinates are rotated to line them up with north-south and east-west, and a distance measure used whose spatial component allows for the presence of rectangular fields. The spectral component uses means and variance but no covariances.

The major differences between the new version of BLOB and the previous one are as follows.

1. A bug was removed from the screening test, a test based on line and point coordinates to quickly remove from consideration many of the current blobs. Also, the new version eliminates line screening, which is not effective.
2. The present version does not have a subset of channels feature, but instead relies on the subset option in POINT or the module SUBCHN. This means that it is no longer possible to blob on a subset of channels and at the same time pass through all the data channels. This option could be put in again without much difficulty if needed.
3. Whereas the previous version updated variances, the present version does not, except that line and point variances can be increased by updating if the control variable BIGGER = 1B.
4. The present version puts out blob means on cards, as did the previous version. However, the most useful form of blob mean output is a tape of blob means in multispectral format put out by STRIP 3.1.
5. The spatial component of the distance measure in the old version was

$$\sqrt{\frac{(\ell - \bar{\ell})^4}{(\text{var } \ell)^2} + \frac{(p - \bar{p})^4}{(\text{var } p)^2}}$$

This measure was an attempt to encourage rectangular, rather than ellipsoidal fields. An equidistant contour is a rectangular-looking super-ellipse like a TV screen.

The new version uses

$$\max \left[\frac{(\ell - \bar{\ell})^2}{\text{var } \ell}, \frac{(p - \bar{p})^2}{\text{var } p} \right]$$



which is faster and has an equidistant contour that is a true rectangle.

6. The new version uses a method of indexing blobs that keeps on the active list only those blobs that are being used. The method is to set a variable SKIP (default 2), and when SKIP lines go by without a blob being chosen by a pixel, the blob is considered completed and retired from the active list. Time is saved by considering only active blobs, and blobs that are long in the column direction are not cut off arbitrarily before they are completed.

HOW TO USE:

BLOB is a POINT module with control input in Step 1. A typical deck set-up is like this:

```
ID Card
$EXECUTE, DUMP, COMPILE MADE
    POINT.(BLOB.)
    E'M
BLOB deck
$DATA
Blob control input
PROCESS input specifying tapes and rectangle or file to be processed
Next rectangle or file
....
MODEL=$START$*
Blob control input with some control values changed
Rectangle to be processed, etc.
```

The BLOB control input is in READ AND PRINT DATA format as follows:

<u>Variable</u>	<u>Default</u>	<u>Description</u>
NCHAN	4	Number of spectral channels used in the blobbing.
VAR	10.	A vector of fixed variances of the spectral data values used as divisors in the distance function. Example: VAR(1) = 36., 9., 36., 9., 36., 9., 36., 9., 9.

<u>Variable</u>	<u>Default</u>	<u>Description</u>
VARL	10.	Variance of the line value.
VARP	10.	Variance of the point value.
BIGGER	0B	When BIGGER = 1B, line and point variances are updated, but only if the updating produces a bigger variance than before.
TAU	9.	When the distance $\sum_1^{NCHAN} \frac{(x_i - \mu_i)^2}{VAR_i} + \max \left[\frac{(\bar{\ell} - \bar{\ell})^2}{VARL}, \frac{(p - \bar{p})^2}{VARP} \right]$ from the pixel $x_1, \dots, x_{NCHAN}$, ℓ, p to the nearest blob mean $\mu_1, \dots, \mu_{NCHAN}$, $\bar{\ell}, \bar{p}$ is less than TAU, the pixel is assigned to that blob and the mean updated. Otherwise the pixel becomes the first member of a new blob. A later paragraph will discuss setting TAU, VAR, VARL and VARP.
MAXCEL	200	The maximum number of blobs on the active list. It does not make BLOB run any faster make MAXCEL small, because only those blobs being used are retained on the active list. If MAXCEL is too small, some pixels will not be blobbed because no more entries are permitted on the active list. (A message will be printed in that case.) At the end of the run, the maximum number of blobs on the active list at one time will be printed. The best plan is to make MAXCEL as big as possible without running out of storage.

<u>Variable</u>	<u>Default</u>	<u>Description</u>
PRINT	1B	A list of blob means is printed (if PRINT = 1B) and/or punched (if PUNCH = 1B) for all blobs except those with fewer than CUT pixels. When WHERE $\neq$ 5, the punched card images are written on tape WHERE.
PUNCH	0B	
CUT	0	
WHERE	5	
EXTRACHAN	0	The blob number is stored in two channels on the output tape, the first containing the integer value of the blob number/512 and the second containing the remainder. If EXTRACHAN (known as RACHAN in the program) is 0, the two channels are put out after the input channels in the output data vector. But if EXTRACHAN is set = 7, for example, then the blob number is stored in channels 6 and 7.
SKIP	2	If SKIP lines go by without a given blob acquiring a pixel, then the blob is considered completed. It is printed and/or punched and retired from the active list.
BADCH	0	When BADCH = 0, you rely on the QCAMS convention that a word QBAD in the title block designates a channel to flag bad data, but when QBAD is 0, then no flagging of bad data is done. If the BLOB user sets BADCH to some number > 0, then that channel is designated as the bad data flag overriding the QCAMS convention.
OLDBAD	0B	When OLDBAD = 1B, you assume the old QCAMS convention that QBAD > 0 and DATUM (QBAD) $\neq$ 0 means that the pixel is bad data. In that case, the blob number is set to 0 and no further processing is done on that pixel.

<u>Variable</u>	<u>Default</u>	<u>Description</u>
		When OLDBAD stays at its default value 0B, then it is assumed that the flag channel was produced by SCREEN and ADDEFLG. Then the decision rule is that
		1. if the only flagging on any pass is for water, the pixel is passed on to further processing
		2. if any pass is flagged cloud, the pixel is assigned to Blob 1 without further processing
		3. otherwise the pixel is assigned to Blob 0 without further processing.

A problem in using BLOB is the setting of the BLOB parameters VAR(1)...VAR(NCHAN), VARL, VARP and TAU. These values are relative; if all are multiplied by 2, say, the decision rule is unchanged. The size of TAU controls the number of blobs. If TAU is too small, a great many small blobs are created. If TAU is too large, then extraneous data is glued together in large blobs. The relative size of VARL, VARP and VAR determines the relative weight given to spatial, as opposed to spectral, homogeneity within blobs. In other words, values of VARL and VARP that are too large produce blobs that are like spectral clusters, scattering themselves all over the map, picking up pixels that are spectrally similar. If VARL and VARP are too small, then neighboring pixels will be grouped together with little regard for spectral similarity. The blobs will be nicely contiguous, but they won't in general correspond to a pattern of fields characterized by spectral homogeneity.

In BLOB runs on Kansas blind site data, R. Kauth and I determined blob parameters as follows. The channels chosen for blobbing were the Tasselled Cap channels brightness and green for four biophases and the yellow channel for the fourth biophase -- nine channels in all. The values of VAR(1)...VAR(9) were set by R. Kauth at 36., 9., 36., 9., 36., 9., 36., 9., 9. The basis of this guesstimate was a belief that a residual error variation arising from such causes as changes in planting density and atmospheric differences not completely corrected by XSTAR4 would produce a standard deviation of at least 3 units in each channel. More variation would be expected in the brightness direction because of the effect view

angle has on brightness and because of differences in soil brightness. So the brightness standard deviation was guessed to be 6. The important guess is that brightness has twice the standard deviation of the other channels; the absolute values chosen are immaterial.

We next assumed that $VARP = VARL$. So that left two undetermined parameters, $VARL$ and TAU . We then chose one of the blind sites (1041) and made a number of BLOB runs with different values of $VARL$ and TAU . The first set of runs showed us the sensitive ranges to be explored with finer resolution in the second set of runs. We made a stripped blob map of each run and chose the parameter set that produced the best-looking map as judged by size of blobs, absence of large areas where the blobs were stripped away to nothing, and a discernible pattern of roads cutting up the landscape into fields.

The values we ended up with were $VARL = 6$. and $TAU = 30$. This setting produced 573 blobs for site 1041 (117 lines, 196 points) and a maximum of 55 blobs on the active list. The same parameter setting was used for 40 pass combinations from 28 sites. The number of blobs varied from 297 to 1537 and the maximum number on the active list from 44 to 130.

APPENDIX II.2
USER DOCUMENTATION OF STRIP
(W. Richardson)

ROUTINE: STRIP.

VERSION: 3.1

DATE: March 1977

PROGRAMMER: W. Richardson W.R.

LANGUAGE: MAD

SYSTEM: 7094/UMES

INTERFACE: A POINT module to be used after BLOB 3.0

PURPOSE: To distinguish boundary pixels of blobs (i.e., strip the blobs) and to put out a tape of blob means.

COMMENTS: The part of the program that distinguishes boundary pixels closely follows version 2.0 of STRIP written by J. Gleason and A. Pentland with exceptions noted in the "Description" section of this memo. The part of the program that puts out the mean tape is new.

SUBROUTINES
NEEDED: Subroutines QAK and SWAP are needed in addition to library routines.

DESCRIPTION:

STRIP puts out two tapes: the pixel tape and the mean tape. The pixel tape is the one put out by the POINT system, except for the last line, which is put out outside the system. It contains all the input channels plus an additional channel (the strip channel) which is 0 if the pixel adjoins, horizontally or vertically, a pixel from another blob, and is 1 otherwise. By mapping the 1's from this channel, we get a picture of the blobs with all the boundary pixels stripped away. If STRIP and MAPP are linked together in one call to POINT, the last line will be missing.

The mean tape contains the blob means and associated information. It is in multispectral format with an arbitrary line length of 120 pixels, each pixel representing a blob. The last line is brought to standard length by filling the unused spaces with zeroes. The first NDATA channels of each pixel are the blob mean vector computed from the interior pixels (i.e., those not identified as boundary pixels) if there are any; if not, the mean is computed from all the pixels in the blob. The additional channels are

<u>Channel</u>	<u>Description</u>
NDATA+1	Line mean.
NDATA+2	Point mean.
NDATA+3	} Blob number in two channels because it may be > 511. Channel NDATA+3 is the quotient after dividing the blob number by 512 and channel NDATA+4 is the remainder.
NDATA+4	
NDATA+5	} Number of pixels in the blob put in two channels as with blob number. I have not observed an example where channel NDATA+5 is greater than zero.
NDATA+6	
NDATA+7	Number of interior pixels in the blob. If this number should happen to be greater than 511, it would be set equal to 511 when put out on tape.
NDATA+8	} Segment number and try number. The try number refers to the set of passes being processed. The segment number (e.g., 1035) and try number (e.g., 1) are put together into one number (e.g., 10351) and then put into two channels as with blob number.
NDATA+9	
NDATA+10	(Optional) Segment index number.

In addition to the generation of a mean output tape, the differences between versions 3.1 and 2.0 of STRIP include

1. A bug causing incorrect results for the first line is removed.
2. Bad pixels, identified by a blob number of 0, get a 0 in the strip channel but adjacent pixels are not stripped. Version 2.0 had a line-numbering problem in this case.
3. A provision is made to put out the last line on tape. This feature works only for tape output; if the output is a DATUM vector to a subsequent module, the last line will be incorrect.

4. The scheme for indexing active blobs is the same as for BLOB 3.0; therefore, STRIP 3.1 should be used on BLOB 3.0 output.

HOW TO USE:

STRIP is a POINT module with control input in Step 1. As has been said before, it is used correctly only on the output of BLOB 3.0. The STRIP pixel output is correct for the last line only if the output goes directly on tape rather than to a subsequent module. On both output tapes, one field is produced for each rectangle or file of data processed. Don't forget to include subroutines QAK and SWAP in the job deck. The control input is in READ AND PRINT DATA format as follows:

<u>Variable</u>	<u>Default</u>	<u>Description</u>
NDATA	0	The number of spectral channels of data coming in, not including flags, blob number, etc. When NDATA is left at its default value 0, the number of spectral channels is set to QNDAT, a word in the title record.
TAB	1B	When TAB = 1B, a mean output tape is printed. When TAB = 0B, the calculation and printing of the means is bypassed.
MBIN	0	} When TAB = 1B, MBIN must be set to the bin number of a tape or to \$REWIND\$, and MUNIT set to a non-conflicting tape unit. (Unit 3 is usually free.)
MUNIT	0	
MFILE	1	Output file on the mean tape. If two files of blob means have already been written on the mean tape, for example, you can start a job with MFILE = 3 and have all three files on the tape with file numbers 1, 2, and 3.
PRINT	0B	IF PRINT = 1B, two lines of information are printed for each blob:



<u>Variable</u>	<u>Default</u>	<u>Description</u>
PRINT (Continued)		1. blob number, number of pixels in the blob, number of interior pixels in the blob, the mean vector computed from all pixels in the blob 2. line mean, point mean, the mean of the interior pixels in the blob.
MISS	2	If MISS lines go by without any pixels from a certain blob on the active list, then that blob is considered completed. It is printed (if PRINT = 1B), stored in the mean output tape buffer and retired from the active list. MISS is the same variable as SKIP in BLOB 3.0 and must have the same value or errors will occur.
MAXCEL	200	The maximum number of blobs on the active list. MAXCEL must be larger than the maximum length of the active list in the preceding BLOB run or errors will occur. This number occurs in the BLOB printout with the message "MAXIMUM NUMBER OF BLOBS ON A LINE =". It does not save time to make MAXCEL small, only space.
BCHAN BCHAN2	0 0	} When these variables are left at 0, it is assumed that the blob number is sorted in the last two channels of the input data vector. If these variables are given positive integer values, then the quotient after dividing the blob number by 512 is assumed to be found in channel BCHAN2 and the remainder in channel BCHAN. Thus BCHAN2 is the first of the two channels containing the blob number.



FORMERLY WILLOW RUN LABORATORIES, THE UNIVERSITY OF MICHIGAN

B120	120	The mean output tape has an artificial line length of B120. Normally, this variable would not be mentioned and the line length left at 120.
ENOUGH	1	The blob mean putout on the mean tape is normally the mean of the interior pixels if there are any, and otherwise is the mean of all the pixels in the blob. But you can set ENOUGH to a number > 1 and then the interior pixels are used only if there are at least ENOUGH of them.
OUT8	08	When data from the mean output tape is going to be used by R. Hieber's program 17RAND, you set OUT8 = 1B, which has the effect of making the number of channels a multiple of 8. The unused channels are filled by zeroes.

The following control variables are read in Process input:

ID	blank	ID can be set with a segment and try number as ID(1) and ID(2) or the last two values of the NSA vector. If left blank, STRIP will take the segment number from the title record and assign a try number of 0.
ICHAN	0	When ICHAN is given an integer value in Process input, that value is stored in channel NDATA+10 of the mean output tape. It provides a quick way of referencing the segment try.

APPENDIX II.3
USER DOCUMENTATION OF BCLUST
(W. Richardson)

ROUTINE: BCLUST
VERSION: 1.0
DATE: November 1977
PROGRAMMER: W. Richardson
LANGUAGE: XTRAN
SYSTEM: Amdahl, MTS
INTERFACE: An 11-line module to be used after 17RAND.
PURPOSE: To group blobs from many segments into spectrally similar strata. The clustering may include ancillary, as well as spectral variables.

SUBROUTINES NEEDED:

Subroutine RANKP is needed in addition to the standard 11-line subroutines.

BCLUST reads control input in NAMELIST format. Variables not mentioned remain at their default values. It also reads a table of ancillary variable data, if requested, and a transformation EV intended to make the distributions representing different materials and varieties approximately spherical.

The blobs are clustered by assigning each blob, in turn, to the cluster it is nearest to or to let it be the start of a new cluster if the smallest distance is greater than a parameter TAU. The EV transformation allows Euclidean distance to be used. The clustering can be iterated by starting a subsequent pass with clusters obtained on the preceding pass. How much weight to give the clusters on the previous pass is controlled by a variable UPFAC. An iteration may be made in two steps by writing the cluster means and sizes into a file and then reading that file to seed clusters on the subsequent pass.

The input tape or file is in multispectral format but with each blob playing the role of a pixel. The first channels are the blob mean. Other channels give the line mean, the point mean, the blob number, the number of pixels in the blob, the number of pixels in the blob after

stripping the boundary pixels, and the segment number. The output is the same as the input except that a channel containing the cluster number is added.

The control variables and their default values are as follows:

<u>Variable</u>	<u>Default</u>	<u>Meaning</u>
NSEG	40	Number of rows to be read from the ancillary data file. Channel 19 on the blob input tape (or file) indicates which row of the ancillary data table is relevant.
NDAT	9	Number of spectral data channels to be used.
NAV	15	Number of ancillary variables to be used.
NY	26	Number of EV-transformed channel values to be used.
MAXCEL	100	Maximum number of B clusters permitted. This number cannot be increased without recompiling the program with an increased value of the literal DM in line 3.
TAU	10.	Distance limit determining whether a blob should start a new B cluster.
BCHAN	0	Channel value the B cluster number will be stored in. If left at the default value, the B cluster channel number will be set to the number of input channels + 1.
NMIN	3	Minimum number of blobs in a B cluster to permit iteration or seeding of that cluster.
UPFAC	1.	If you are seeding or iterating, UPFAC determines how many blobs you pretend are already in the seeded cluster. If UPFAC = 1., the pretend number is the actual number of the preceding iteration. If UPFAC = 0., the pretend number is 1. UPFAC > 0 and < 1 gives a number that is between 1 and the number in the cluster proportionately. UPFAC > 1 puts even greater weight on the number already in the cluster.

CREATE	T	If CREATE is T (short for .TRUE.) then a new cluster is created whenever the minimum distance is $\geq$ TAU.
UPDATE	T	When UPDATE is T, then the mean of a cluster is updated every time a blob is included in the cluster. If no seeding or iterating has been done, the updating has the effect of placing the cluster mean at the arithmetic mean of the data values of the blobs in the clusters. When seeding or iterating is done, then it is possible to endow each cluster with an artificial number of points which have the effect of making the means less moveable.
TAUFAC	0	On iteration, it is possible to bring the number of B clusters toward a chosen number GOODNO of clusters. When TAUFAC = 1. $\text{new TAU} = \left(\frac{\text{GOODNO}}{\text{No. of clusters on preceding pass}} \right) \text{old TAU}$ When TAUFAC = 0., no such automatic changing of TAU takes place. TAUFAC between 0 and 1 makes a proportionate change between these two extremes. This option applies to iteration only, not seeding.
GOODNO	60	
NEWANC	T	IF NEWANC = F, then no ancillary data is read. NEWANC is set = F if NAV = 0. This option would be used on a second pass when the ancillary data had already been read into the ancillary variable table.
CHAN	1, 2, 3, ...	CHAN (1) ... CHAN(NDAT) is the subset of data channels to be used in the B clustering.
CHANA	1, 2, 3, ...	CHANA(1) ... CHANA(NAV) is the subset of ancillary variables to be used in the B clustering. The ancillary variable are considered to be numbered as follows: 1 longitude 2 latitude 3 elevation 4-7 four Julian dates 8-11 four view angles 12-15 four gammas

ITERAT F You set ITERAT = T for a subsequent pass on which you want to set the cluster means at the values of the previous pass. Only those clusters with NMIN or more blobs are used. For updating purposes, each such mean is associated with an artificial number of points ranging from 1 (UPFAC = 0.) to the number in the cluster on the previous pass (UPFAC = 1.) [A problem with using ITERAT is that you generally don't want an output tape on the first pass but do on the second. I have not tested out ways of overcoming this problem.]

SEED F When SEED = T, then it is assumed that you are running a subsequent pass and that the number of points in each cluster and its mean vector has been saved in a file which is read by the I/O assignment 16. The saving on the previous pass is done by writing into I/O assignment 17. The previous comments on NMIN and UPFAC apply, but the TAUFAC option does not.

READEV T READEV = T means read an NxN matrix EV from I/O assignment 14 for the purpose of transforming the data to a space where the distributions are approximately spherical. N is NDAT + NAV. On a subsequent pass using ITERAT, you should turn READEV off or the program will hang up.

 If READEV = F, the EV matrix is not read. An EV matrix previously read would then be left unchanged. The default is the identity matrix.

IT { 1, 1, . . . IT(I) = 0 means don't use this segment.
NIT } 19 I refers to a segment number that is found in channel NIT (presently 19). This channel is not the usual segment-try number (e.g. 10201) but rather an index number between 1 and 40 that is used not only for the IT option but also to access the row in the ancillary variable table to be used.

DEBUG F When DEBUG is T, debugging printout is written for every blob. When DEBUG = T, you generally specify a very small number of blobs to be processed.



The input-output assignments are as follows. 4 = *SOURCE*, 6 = *SINK*, 8 = *PRINT*, 11 = the multispectral input tape or file and 10 = the multispectral output tape or file, as is standard for 11-line modules.

The ancillary data table is read from 15 if NEWANC is left at its default value of T. The successive lines of 15 are read unformatted and put into the rows of the table. The row number corresponds to the segment index number in channel 19 of the input, so that the table can be accessed efficiently.

The EV transformation to make the distributions approximately spherical is read from unit 14. It is assumed to be a square matrix stored row by row on one line of 14 and read unformatted. The size of the matrix is NCHAN by NCHAN, where NCHAN is the total number NDAT + NAV of variables. It defaults to the identity matrix.

If the SEED option is used, cluster means are written unformatted on unit 17 at the end of one pass and then read unformatted on 16 on the next pass.

APPENDIX II.4

ANNOTATED LISTING OF WCT

(W. Holsztynski)

WCT is a program to select $Q(1) + Q(2) + \dots$ segments to represent (NH-NDEL) segments. The main program reads an incidence matrix, IM, which is the number of blobs in each spectral stratum/segment combination.

Subroutine GET01 converts the incidence matrix to an array of zeros and ones, where a zero is inserted whenever the number of blobs is less than LB, the "lower bound".

Subroutine KRS computes the value of various segment choices. Suppose 4 segments are to be chosen in groups of two. The main program calls KRS which first picks the best pair and returns to the main program. The main program again calls KRS which pick the best complementary pair and returns.

Subroutine NEXKSB is called by KRS and provides KRS with all combinations of available segments taken 2 at a time (in the above example).

After all segments have been chosen, the main program calls subroutine FINCH to make the final choice of training blobs within the training segments. The total number of blobs to be chosen may be set, (NTB), or the proportion of them (NPR). The choices are distributed proportionally among the spectral strata and further distributed among the training segments. In the particular version listed here the distribution of blob choices depends upon the number of pixels in each spectral stratum rather than upon the number of blobs, and so the program also reads a pixel array, PIX.

Control data is entered in NAMELIST format when prompted by the program. The listing follows on the following four pages.



ORIGINAL PAGE IS
OF POOR QUALITY

FORMERLY WILLOW RUN LABORATORIES, THE UNIVERSITY OF MICHIGAN

```
1  EMPTY NCT,0
2  $RUN *FTN 1=2 PAR=SE+SOURCE* (NCT,0 P=PR[NT] TEST ERR
3  IMPLICIT INTEGER(A-Z)
4  C A TYPICAL I/O LIST IS 1=TAGSPNA13 3=PIXSPNA13 5=SOURCE* 6=SI[NK* 8=H3PNA13862
5  LOGICAL MTC,GAP
6  DIMENSION IM(100,60),M(100,60),EM(60),EW(16),TT(40),
7  IR(16),X(16),MI(100),C(60),O(16),CH(16),MR(100,60),TD(40)
7.2  I,PIX(100,60)
8  DATA Fw /100,20,4,13*0/,NTB,NPR/300,0/,
9  IQ/1,15*0/,LB/12/,NH,NV/40,100/, TD/40*0/, NDEL/0/
10  NAMELIST/RODATA/TD,NDEL,Fw,NV,NH,LB,0,NTB,NPR
11  VAL=0
12  UP=0
13  GAP=.FALSE.
14  DO 40 I=1,15
15  IF (Fw(I).EQ.0,AND,EW(I+1),NE.0) GAP=.TRUE.
16  40  CONTINUE
17  REWIND 1
18  REWIND 3
19  READ(3,2) NV,NH
20  READ(1,2) NV, NH
21  WRITE(6,6)
22  6  FORMAT(' NAMELIST,RODATA ')
23  READ(5,RODATA)
24  DO 7 I=1, 40
25  7  TT(I) = 1
26  IF (NDEL.FD.0) GO TO 9
27  DO 8 I=1,NDEL
28  IF (TD(I).GE.1,AND, TD(I).LE.40) GO TO 8
29  WRITE (6,10) NDEL, TD
30  10  FORMAT (' NDEL =', 15, ', TD = ', 10/10/ (20x, 10/10) )
31  STOP
32  8  TT(TD(I)) = 0
33  9  WRITE (6,RODATA)
34  2  FORMAT (16/5)
35  DO 56 I=1,NV
36  56  ML(I)=0
37  DO 3 I=1,NH
38  3  C(I) = 0
39  Fw(I)=1
40
41  READ(1,2)((IM(I,J),J=1,NH),I=1,NV)
42  READ(3,2)((PIX(I,J),J=1,NH),I=1,NV)
43  DO 4 J=1,NH
44  DO 4 I=1,NV
45  MI(I,J)=IM(I,J)
46  4  CONTINUE
47  CALL GETOI(4,NV,NH,LB,TT)
48  DO 401 J=1,NH
49  DO 401 I=1,NV
50  401  MR(I,J)=M(I,J)
51
52  JC=0
53
54  11  W(1)=Fw(1)
55  DO 5 I=2,16
56  W(I)=W(I-1)+EW(I)
57  5  CONTINUE
58
59  H=NH
```

```

60      VENV
61
62      DO 10 I=1,16
63      IF(Q(I),EQ,0)GO TO 10
64      K=Q(I)
65      VL=VAL
66      CALL KHS(K,H,V,ML,MR,B,VL,W)
67      WRITE(6,44)VL
68      IF(VL,NE,VL,OR,GAP)GOTO 14
69      WRITE(6,13)VAL
70      13 FORMAT(' NO MORE COL-S OF ANY VALUE, VAL=',I10)
71      GOTO 52
72      14 DO 15 J=1,K
73      JC=JC+1
74      CH(JC)=EN(B(J))
75      15 C(CH(JC))=1
76
77      L1=0
78      DO 20 L=1,H
79      IF(C(FN(L)),NE,0) GO TO 20
80      L1=L+1
81      EN(L1)=FN(L)
82      20 CONTINUE
83      N=H=K
84      DO 25 J=1,H
85      DO 25 L=1,V
86      25 MR(L,J)=M(L,EN(J))
87      VAL=VL
88      10 CONTINUE
89      32 IF(JC,EQ,0)GOTO 33
90      WRITE(6,21)(CH(I),I=1,JC)
91      WRITE(6,44)VAL
92      44 FORMAT(' VAL=',I10)
93      GOTO 27
94      33 WRITE(6,34)
95      34 FORMAT(' ALL IM LESS THAN LR')
96      GO TO 100
97
98      27 CALL FINCH(PIX,IM,NV,NH,CH,JC,NPH,NTS)
99      GO TO 100
100     END
101
102     SUBROUTINE GETOI(IM,NV,NH,LR,TT)
103     DIMENSION IM(100,60)
104     INTEGER TT(40)
105     DO 10 J=1,NH
106     DO 20 I=1,NV
107     IF(IM(I,J),LT,LR,OR, TT(J),EQ, 0) GOTO 5
108     IM(I,J)=1
109     GO TO 20
110     5 IM(I,J)=0
111     20 CONTINUE
112     10 CONTINUE
113     RETURN
114     END
115
116     SUBROUTINE KHS(K,H,V,ML,M,B,VL,W)
117     INTEGER H,V,ML(100),M(100,60),B(K),VL,
118     1A(16),M(16),ML(100),MP(100),V1
119     LOGICAL MTC,MAD

```

ORIGINAL PAGE IS
OF POOR QUALITY



FORMERLY WILLOW RUN LABORATORIES, THE UNIVERSITY OF MICHIGAN

```
1  SFMPTY NCT,0
2  SRUN 4PTX T=2 PAR=S2=SHINCE* L=NCT,0 P=PRINT* TEST ERR
3  IMPLICIT INTEGER(A-Z)
4  C A TYPICAL I/O LIST IS 1=TAGSPNA13 3=PIV3PNA13 5=STURCF* 6=SI*NM* 8=3PNA13562
5  LOGICAL MTC,GAP
6  DIMENSION IM(100,60),M(100,60),EM(60),EM(16),TT(40),
7  IR(16),W(16),MI(100),C(60),D(16),CM(16),MR(100,60),TD(40)
7.2  I,PIV(100,60)
8  DATA F* /100,20,4,13*0/,NTB,NPH/300,0/,
9  IQ/1,15*0/,LR/12/,NH,NV/40,100/, TD/40*0/, NDEL/0/
10  NAMELIST/RODATA/TD,NDEL,FM,NV,NH,LR,0,NTB,NPH
11  VAL=0
12  UP=0
13  GAP=.FALSE.
14  DO 40 I=1,15
15  IF (F*(I).EQ.0.AND.EM(I+1),NE,0) GAP=.TRUE.
16  40 CONTINUE
17  READ(10) I
18  READ(10) 3
19  READ(3,2) NV,NH
20  READ(1,2) NV, NH
21  WRITE(6,6)
22  6 FORMAT(' NAMELIST,RODATA ')
23  READ(5,RODATA)
24  DO 7 I=1, 40
25  7 TT(I) = 1
26  IF (NDEL.EQ.0) GO TO 9
27  DO 8 I=1, NDEL
28  IF (TD(I).GE.1 .AND. TD(I).LE.40) GO TO 8
29  WRITE (6,101) NDEL, TD
30  101 FORMAT (' NDEL =', I5, ', TD = ', I10/ (20X, 10110) )
31  STOP
32  8 TT(TD(I)) = 0
33  9 WRITE (6,RODATA)
34  2 FORMAT(16I5)
35  DO 56 J=1,NV
36  56 ML(I)=0
37  DO 3 I=1,NH
38  C(I) = 0
39  3 E*(I)=1
40
41  READ(1,2)((IM(I,J),J=1,NH),I=1,NV)
42  READ(3,2)((PIV(I,J),J=1,NH),I=1,NV)
43  DO 4 J=1,NH
44  DO 4 I=1,NV
45  M(I,J)=IM(I,J)
46  CM(I)=M(I)
47  CALL GETUT(M,NV,NH,LR,TT)
48  DO 401 J=1,NH
49  DO 401 I=1,NV
50  401 MR(I,J)=M(I,J)
51
52  JC=0
53
54  11 M(I)=E*(I)
55  DO 5 I=2,16
56  M(I)=M(I-1)+EM(I)
57  5 CONTINUE
58
59  H=NH
```

```

120      BAD=.TRUE.
121      3  DO 4 I=1,V
122      4  M1(I)=ML(I)
123      V1=0
124      CALL NEYKSB(H,K,A,MTC)
125      DO 5 I=1,V
126      DO 10 J=1,K
127      NA=A(J)
128      10  M1(I)=M1(I)+M(I,NA)
129      IF(M1(I).EQ.0)GOTO 5
130      V1=V1+M1(I)
131      5  CONTINUE
132      IF(K.NE.1)GOTO 17
133      WRITE(6,30)V1,(M1(I),I=1,V),A(1)
134      30  FORMAT(' V1=',I5,' M1:',34I1,' A:',I2)
135      17  IF(V1.LE.VL)GOTO 21
136      BAD=.FALSE.
137      DO 15 I=1,K
138      15  M(I)=A(I)
139      VL=V1
140      DO 20 I=1,V
141      20  M2(I)=M1(I)
142      21  IF(MTC)GOTO 3
143      IF(BAD) RETURN
144      DO 25 I=1,V
145      25  ML(I)=M2(I)
146      RETURN
147      END
148
149      SUBROUTINE NEYKSB(N,K,A,MTC)
150      INTEGER A(K), H
151      LOGICAL MTC
152      DATA NLAST/0/,KLAST/0/
153      10  IF(K.NE.KLAST.(OR.N.NE.NLAST))GO TO 20
154      30  IF(MTC) GO TO 40
155      20  M2=0
156      H=K
157      NLAST=N
158      KLAST=K
159      MTC=.TRUE.
160      GO TO 50
161      40  DO 41 M=1,K
162      M2=A(K+1-M)
163      IF(M2.NE.N+1-M) GO TO 50
164      41  CONTINUE
165      50  DO 51 J=1,H
166      51  A(K+J-M)=M2+J
167      MTC=(A(1).NE.N-K+1)
168      RETURN
169      END
170
171      SUBROUTINE FINCH(PIX,IM,NV,NH,CH,JC,NPR,NB2)
172      [INTEGER IM(100,60),CH(16),T,D(100,16),RT(100),NDT, AL(100),PIX(100
172.5  C,60), PT(100)
173      NTH = NB2
174      NTH = 1
175      TE=0
176      TP=0
177      DO 1 I=1,NV
178      HT(I)=0

```

```

179      PT(I)=0
180      DO 2 J=1,NH
181      RT(I)=RT(I)+IM(I,J)
182      PT(I)=PT(I)+PIX(I,J)
183      2 CONTINUE
184      TP=TP+PT(I)
185      T=T+RT(I)
186      1 CONTINUE
187      IF(NPR.NE.0) NTR=1+(NPR*1)/1000
188      10 DO 3 I=1,NV
189      AL(I)=(TP/2+NTR*PT(I))/TP
190      3 CONTINUE
191      200 DO 4 I=1,NV
192      DO 4 J=1,JC
193      4 D(I,J)=0
194      NDT=0
195      ND=0
196
197      DO 5 I=1,NV
198      J=MOD(I,JC)
199      IF(J.EQ.0) J=JC
200      NDT=NDT+ND
201      ND=0
202      NF=0
203      6 IF(ND.GE.AL(I)) GO TO 5
204      IF(D(I,J).LT.IM(I,CH(J)))GO TO 7
205      NF=NF+1
206      IF(NF.GE.JC) GO TO 5
207      GO TO 8
208      7 D(I,J)=D(I,J)+1
209      ND=ND+1
210      NF=0
211      8 J=MOD(J+1,JC)
212      IF(J.EQ.0) J=JC
213      GO TO 6
214      5 CONTINUE
215      IF(IARS(NTB2-NDT).LF.10 .OR. NTIMES.GT.30) GO TO 35
216      NTIMES = NTIMES + 1
217      NTR = NTR + (NTR2 - NDT)
218      GO TO 10
219
220      35 WRITE(8,80)JC,NV,(CH(J),J=1,JC)
221      80 FORMAT(2(1X,I2),5(5(1X,I3)/))
222      WRITE(6,40)(CH(J),J=1,JC)
223      40 FORMAT(4X,'WRITE',15,1517)
224      WRITE(6,50)
225      50 FORMAT(1X,'R=CL',2X,16(2X,'IM,D'))
226      DO 70 I=1,NV
227      WRITE(6,60)I,(IM(I,CH(J)),D(I,J),J=1,JC)
228      60 FORMAT(14,3X,16(14,' ',I2))
229      WRITE(8,90)(D(I,J),J=1,JC)
230      70 CONTINUE
231      90 FORMAT(20(1X,I3))
232      WRITE(6,85)NDT
233      85 FORMAT(' NDT=',I10)
234      RETURN
235      END
236      $ENDFILE
237      $SOURCE PREVIOUS
END OF FILE

```

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX III

A NOTE ON THE RATIONALE FOR PARTITIONING IN LARGE AREA REMOTE SENSING SURVEYS

(R. Kauth and W. Richardson)

In carrying out large area remote sensing surveys the remotely-sensed image data may not, by itself, be sufficiently definitive of the classes or conditions we seek to survey; ancillary information, such as weather reports and prior years' histories will often be required. The remote-sensed image features themselves may be the result of a preprocessing transformation on the original image data, e.g., a haze correcting transformation. How can the ancillary data be used in conjunction with the remote sensing image features to classify a scene?

Suppose we have a disjoint set of recognition classes, $i = 1, \dots, m$ and a discrete random variable w which takes these values. Then the method of classification is to choose that w (i.e., choose that class) whose posterior probability is a maximum given both the feature and ancillary data observations. In symbols,

$$\text{choice} = \max_w \text{prob}(w|y,v),$$

where y is the image feature data and v is the ancillary data. Manipulating this expression by Bayes rule,

$$\begin{aligned} \max_w \text{prob}(w|y,v) &= \max_w \frac{\text{prob}(w,y,v)}{\text{prob}(y,v)} \\ &= \max_w \frac{\text{prob}(y,v|w) \text{prob } w}{\text{prob}(y,v)} \end{aligned}$$

Since for each observation the values of y and v are fixed while the expression is maximized over w , the denominator can be ignored, thus the rule is

$$\max_w \text{prob}(y, v|w) \text{prob } w. \quad (1)$$

This is just the maximum likelihood procedure for y and v taken as a single vector observation with prior weights assigned to the classes. Thus the proper way to use ancillary data is to add ancillary channels to the data, train over a wide range of ancillary conditions, and classify.

The immediate objection to this approach is that our present large scale system structure is not set up to handle these added channels in the mainstream processor. In addition, the training function would have to be organized on a very large scale. Finally, in order to actually implement this approach, confidence in the fundamental concepts of likelihood model classification is required.

The first of the above problems can be surmounted by a modified form of the maximization problem, thus,

$$\begin{aligned} \max_w \text{prob}(w|y, v) \\ &= \max_w \frac{\text{prob}(w, y, v)}{\text{prob}(y, v)} \\ &= \max_w \frac{\text{prob}(y|v, w) \text{prob}(w|v) \text{prob } v}{\text{prob}(y, v)} \\ &= \max_w \text{prob}(y|v, w) \text{prob}(w|v) \end{aligned} \quad (2)$$

With this formulation one trains the likelihood model for $\text{prob}(y|v, w)$. When processing a segment, the values of v for that segment are used to set the signatures for processing the image vectors, y . The conditional weight must also be trained, but its accuracy is not critical. Thus, in this formulation we have moved the problem of establishing signatures off line and allowed the use of a conventional image processor but the problem of training is just as severe as before.

An approximation to the above procedure can be obtained by partitioning the survey region into bounded domains of the ancillary variables and then training and classifying within each partition. Thus the objective of partitioning is to make a step-wise approximation to the likelihood function

$$\text{prob}(y|w, v) \approx q_k(y|w), v \in R_k$$

where R_k is the region covered by the k^{th} partition.

If the training were completely representative over each partition then,

$$q_k(y|w) = \frac{\int_{R_k} \text{prob}(y|v, w) \text{prob}(v) dv}{\int_{R_k} \text{prob}(v) dv} \quad (3)$$

Also the prior weight can be approximated,

$$\text{prob}(w|v) \approx r_k(w), v \in R_k$$

where

$$r_k(w) = \frac{\int_{R_k} \text{prob}(w|v) \text{prob}(v) dv}{\int_{R_k} \text{prob}(v) dv}$$

Thus the classification rule becomes,

$$\max_w q_k(y|w) r_k(w)$$

The effect of partitioning is to increase the variance of the resulting signatures relative to what it would have been if the more correct formulation, Equation 2, were used.

Thus,

$$\text{Var} (q_k (y|w)) \geq \text{Var} (\text{prob} (y|v_k))$$

where v_k is the value of the ancillary variable at some arbitrary point in the k^{th} partition. The equal sign would be employed only if $\text{prob}(v)$ in Equation 3 were a delta function, $\delta (v-v_k)$, or if the $\text{prob}(y|v,w) = \text{prob} (y|w)$ independent of v . Neither of these conditions is likely in practice, therefore the partition-wide signatures have increased variance and increased classification error results.

An additional source of errors is failure to train sufficiently within the partition, leading in effect to errors in estimate of $\text{prob} (y|v,w)$. These errors can be expected to decrease asymptotically to zero as the number of training samples increases. The number of training samples is limited by the size of the partition; hence there is some partition size at which the sum of errors from the spreading of the likelihood model and the errors from insufficient training is a minimum.

Finally, an additional reason for increasing the size of partitions is to minimize the cost of training, by increasing the ratio of the number of recognition segments to the number of training segments.

The above setting gives us a rationale for partitioning. Partitioning followed by within-partition training and classification is an approximation to a more correct procedure contained in Equation 2. The increased variance of the partition-wide signatures causes increased classification errors, making it desirable to decrease partition size. This is balanced by training errors and cost considerations which make it desirable to increase partition size.

Methodology for Estimating a Metric for Clustering Ancillary Data

A reasonable approach to partitioning is to cluster available segments within the space defined by their ancillary data. In order to

use such a method some reasonable metric is required, i.e., some measure of the relative importance of the various ancillary variables or combinations of them. The metric is expressed in the form of a covariance matrix $\sum_v$ that is used to determine which cluster each v point (i.e., each segment) belongs to. $\sum_v$ characterizes the relative stability of $\text{prob}(y|v,w)$ as v changes in different directions. It will, in general, be quite different from the covariance matrix C_v of the distribution of v .

The problem of partitioning is thus transformed into the problem of finding a reasonable estimate for $\sum_v$. In the case of the wheat survey, this estimate might be based on an estimate of the wheat signature alone or on performance using both the wheat and confusion crop signatures or on a training sufficiency basis. We propose, as a first step, to estimate $\sum_v$ using only the signature for dryland wheat.

Consider a dryland wheat probability density function which is a joint function of both the feature variable y and the ancillary variable v . Assume that the distribution of y and v is joint multivariate normal and that this description is valid over a region within which we expect to create partitions. We let z be the joint vector $\begin{pmatrix} y \\ v \end{pmatrix}$ and C_z be the covariance matrix of z

$$= \begin{pmatrix} C_y & C_{yv} \\ C_{yv}^T & C_v \end{pmatrix}$$

where C_y is the covariance matrix of y and C_v , the covariance matrix of v . We let the mean be zero for the convenience of representing y and v as deviations from the mean. In short, z is normal $(0, C_z)$.

We are, as will be seen, particularly interested in the conditional distribution of y given v . This is

$$\begin{aligned} \text{prob}(y|v) &= \frac{\text{prob}(y,v)}{\text{prob}(v)} = \frac{\text{prob}(z)}{\text{prob}(v)} \\ &= \text{constant} \cdot e^{-\frac{1}{2} A} \end{aligned} \quad (4)$$

We can write the argument, A, as

$$\begin{aligned}
 A &= \mathbf{z}^T \mathbf{C}_z^{-1} \mathbf{z} - \mathbf{v}^T \mathbf{C}_v^{-1} \mathbf{v} \\
 &= \begin{pmatrix} y \\ v \end{pmatrix}^T \begin{pmatrix} C_y & C_{yv} \\ C_{yv}^T & C_v \end{pmatrix}^{-1} \begin{pmatrix} y \\ v \end{pmatrix} - \mathbf{v}^T \mathbf{C}_v^{-1} \mathbf{v}
 \end{aligned} \tag{5}$$

Let

$$\begin{pmatrix} C_y & C_{yv} \\ C_{yv}^T & C_v \end{pmatrix}^{-1} = \begin{pmatrix} Q_y & Q_{yv} \\ Q_{yv}^T & Q_v \end{pmatrix}$$

Then,

$$\begin{aligned}
 A &= \begin{pmatrix} y \\ v \end{pmatrix}^T \begin{pmatrix} Q_y & Q_{yv} \\ Q_{yv}^T & Q_v \end{pmatrix} \begin{pmatrix} y \\ v \end{pmatrix} - \mathbf{v}^T \mathbf{C}_v^{-1} \mathbf{v} \\
 &= \left(y^T Q_y + v^T Q_{yv}^T, y^T Q_{yv} + v^T Q_v \right) \begin{pmatrix} y \\ v \end{pmatrix} - \mathbf{v}^T \mathbf{C}_v^{-1} \mathbf{v} \\
 &= y^T Q_y y + v^T Q_{yv}^T y + y^T Q_{yv} v + v^T Q_v v - \mathbf{v}^T \mathbf{C}_v^{-1} \mathbf{v} \\
 &= y^T Q_y y + 2y^T Q_{yv} v + v^T (Q_v - \mathbf{C}_v^{-1}) v
 \end{aligned}$$

By the Roy and Sarhan theorem,

$$C_v^{-1} = Q_v - Q_{yv}^T Q_y^{-1} Q_{yv}$$

$$A = y^T Q_y y + 2y^T Q_{yv} v + v^T (Q_{yv}^T Q_y^{-1} Q_{yv}) v$$

Let

$$Q_y = K_y^T K_y \quad (\text{Cholesky decomposition})$$

$$Q_y^{-1} = K_y^{-1} K_y^{-1T}$$

$$\begin{aligned} A &= y^T K_y^T K_y y + 2y^T Q_{yv} v + v^T Q_{yv}^T K_y^{-1} K_y^{-1T} Q_{yv} v \\ &= (K_y y + K_y^{-1T} Q_{yv} v)^T (K_y y + K_y^{-1T} Q_{yv} v) \\ &= [K_y (y + K_y^{-1} K_y^{-1T} Q_{yv} v)]^T [K_y (y + K_y^{-1} K_y^{-1T} Q_{yv} v)] \\ &= (y + Q_y^{-1} Q_{yv} v)^T Q_y (y + Q_y^{-1} Q_{yv} v) \end{aligned} \quad (6)$$

Thus $\text{prob}(y|v)$ has a mean of $-Q_y^{-1} Q_{yv} v$ and a covariance Q_y^{-1} independent of the value of v .

Now we wish to establish a distance measure in the ancillary variable space which will represent the importance of any deviation v from its mean of 0 in causing change in the distribution of y given v . A measure of this sensitivity is the probability of observing the point y at its mean 0. We have

$$\text{prob}(y = 0|v) \sim e^{-\frac{1}{2} (Q_y^{-1} Q_{yv} v)^T Q_y (Q_y^{-1} Q_{yv} v)}$$

and v values for which the argument is constant can be said to represent equally important shifts in v .

$$\begin{aligned} A &= v^T Q_{yv}^T Q_y^{-1} Q_y Q_y^{-1} Q_{yv} v \\ &= v^T (Q_{yv}^T Q_y^{-1} Q_{yv}) v \end{aligned}$$

If we let $\sum_v^{-1} = Q_{yv}^T Q_y^{-1} Q_{yv}$, then

$$A = v^T \sum_v^{-1} v \quad (7)$$

Thus we have found that

$$\text{prob}(y = 0|v) = e^{-\frac{1}{2} v^T \sum_v^{-1} v}$$

This expression can also be viewed as a distribution $g(v)$ in v space reflecting the stability of $\text{prob}(y|v)$ because an equally dense contour of $g(v)$ in v space corresponds to a constant $\text{prob}(y = 0|v)$. $\sum_v$, the covariance matrix of $g(v)$, is thus an appropriate covariance to use in clustering the points of v .

We have shown that if we cluster the segments by clustering their ancillary data values v using the covariance matrix

$$\sum_v \sim (Q_{yv}^T Q_y^{-1} Q_{yv})^{-1} \quad (8)$$

we will give proper relative weight to the ancillary variables in terms of their ability to effect the dryland wheat signature.

In order to carry out the above prescription it is necessary to obtain an approximate description of the covariance of the vector $\begin{pmatrix} y \\ v \end{pmatrix}$

over the entire region which we hope to partition. But it should be born in mind that it is not critically important how accurate this description is, since it is not critically important to be very accurate in choosing partition boundaries. However the matrix $\sum_v$ can tell us some important things. The least eigenvector of this matrix points in the direction of the most critically important ancillary variables. At the opposite extreme the eigenvector whose eigenvalue is largest points to the least important linear combination of ancillary variables.

The size and number of partitions can be controlled by varying the distance limit, a clustering parameter that determines when a point should start a new cluster.

It is quite possible that the unconditional covariance of v , C_v , is singular or nearly so, especially since some of the ancillary variables may be highly correlated to each other. Then, even though $\sum_v$ is not near singular, (being a projection of the feature data, y , onto the ancillary data space) it would not be possible to compute it by Equation 8, since Q_{yv}^T and Q_y will not be calculable, since C_z will be singular. The possibilities for surmounting this difficulty include using a generalized inverse; taking only the best linear combinations of v components; and adding a small fixed noise variance to the matrix C_v before making the calculation.

APPENDIX III.1

HOW TO FIND A METRIC FOR PARTITIONING THE ANCILLARY VARIABLE SPACE
WHEN THE COVARIANCE MATRIX OF THE ANCILLARY VARIABLE IS SINGULAR

(W. Richardson and R. Kauth)

The discussion requires the use of certain theorems about the generalized inverse of a symmetric matrix. These theorems are given in Part 1. The substance of the appendix, a continuation of Appendix III, is given in Part 2.

1. THE GENERALIZED INVERSE OF A REAL SYMMETRIC MATRIX

In this section, the generalized inverse A^+ of a real symmetric matrix A is defined. A method for finding this inverse is summarized in Theorem 1. Two other results that are needed will be proved as Theorems 2 and 3.

Odell and Boullion* define a generalized inverse or "pseudoinverse" A^+ of the complex matrix A as a matrix X satisfying the following four conditions:

$$AXA = A$$

$$XAX = X$$

$$(XA)^T = XA$$

$$(AX)^T = AX$$

If A is real and symmetric, the generalized inverse can be found as follows:

There exists an orthogonal matrix P such that

$$PAP^T = \text{diagonal}$$

* Books discussing the generalized inverse are given as references [6]-[8].

To fix our ideas, let us suppose that the non-zero diagonal elements (eigenvalues) are in the upper left. P is the matrix of the eigenvectors.

$$PAP^T = \begin{bmatrix} \alpha_1 & & & \\ & \dots & & \\ & & \alpha_r & \\ & & & 0 \\ & & & 0 \\ & & & 0 \end{bmatrix}$$

Let us suppose, further, that A is positive semidefinite as it will always be if it is a covariance matrix. Then $\alpha_1, \dots, \alpha_r$ are all > 0 . Let

$$S = \begin{bmatrix} \frac{1}{\sqrt{\alpha_1}} & & & \\ & \dots & & \\ & & \frac{1}{\sqrt{\alpha_r}} & \\ & & & 1 \\ & & & 1 \\ & & & 1 \end{bmatrix}$$

Then

$$SPAP^TS = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \dots & \\ & & & 1 \\ & & & 0 \\ & & & 0 \\ & & & 0 \end{bmatrix} \stackrel{\text{def}}{=} I_r$$

If we let T be the matrix SP and if X has covariance matrix A , then $Y = TX$ has covariance matrix $TAT^T = I_r$.

Odell and Boullion give the following formulation of a matrix satisfying the first condition $AXA = A$. Find non-singular matrices P and Q such that

$$PAQ = I_r$$

Then

$$X = QI_r P$$

satisfies the first condition $AXA = A$.

Proof: To prove: $A QI_r P A = A$
 i.e., to prove: $A Q I_r P A Q = A Q$
 i.e., to prove: $P A Q I_r P A Q = P A Q$
 i.e., to prove: $I_r I_r I_r = I_r$ which it is.

A formal proof takes these steps backward, first premultiplying by P^{-1} then postmultiplying by Q^{-1} . You can go backward or forward in these proofs when the matrices are non-singular.

This same X also satisfies the second condition

$$XAX = X$$

Proof: To Prove: $QI_r P A Q I_r P = QI_r P$

The PAQ in the middle $= I_r$. Hence, we must prove that $QI_r I_r I_r P = QI_r P$, which it does.

For A real, symmetric and positive definite, SP plays the role of P in the above arguments and $P^T S$ plays the role of Q . SP and $P^T S$ are non-singular and $(SP) A (P^T S) = I_r$. Hence, $(P^T S) I_r (SP)$ has been shown to satisfy the first two conditions of A^+ .

We will show that $P^T S I_r SP$ satisfies the last two conditions also and is therefore A^+ . We first observe that A and $X = P^T S I_r SP$ are both symmetric.

$$X^T = P^T S^T I_r S^T P = P^T S^T I_r SP = X$$

because $S^T = S$. Therefore

$$(XA)^T = A^T X^T = AX$$

To prove: $AX = XA$

i.e., $AP^T S I_r SP = P^T S I_r SPA$
 i.e., $(SP)AP^T S I_r SP = (SP)P^T S I_r SPA$

because SP is non-singular

$$\text{i.e., } (SP)AP^T S I_r SP(P^T S) = (SP)P^T S I_r SP A(P^T S)$$

$$\text{i.e., } (SP A P^T S) I_r S (P P^T) S = S (P P^T) S I_r (S P A P^T S)$$

$$\text{i.e., } I_r I_r S^2 = S^2 I_r I_r \text{ which it is. Q.E.D.}$$

Finally, we must prove that

$$(AX)^T = AX$$

$$\text{i.e., } X^T A^T = AX$$

$$\text{i.e., } XA = AX \text{ which is what we have just proved.}$$

Thus, $P^T S I_r SP$ is the unique pseudo-inverse A^+ of A .

You might think that with all these desirable properties, AA^+ would = I_r . But such is not the case. If $AA^+ = I_r$, which I_r would it be? The one with positive values in the first r places? The last r places? A scattered subset of size r ?

If $AA^+ = I_r$, then

$$AA^+ A \text{ would} = I_r A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{r1} & \dots & a_{rn} \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix} \neq A$$

so property 1 would not hold.

If we try to prove $AA^+ = I_r$ by previous methods we would try to prove

$$AP^T S I_r SP^T = I_r$$

$$\text{i.e., } SP A P^T S I_r SP^T = S P I_r$$

$$\text{i.e., } I_r I_r SP = S P I_r$$

The left side is a matrix of the first r rows of SP and zeroes elsewhere. The right side matrix has the first r columns of SP . So the attempted proof fails.

Thus, we conclude that the pseudo inverse of a covariance matrix A (real, symmetric, positive semidefinite) is

$$P^T S I_r SP$$

In words, you obtain the pseudo inverse by finding an orthogonal matrix P such that PAP^T is diagonal, take the reciprocal of the diagonal elements, which we have called

$$SI_r S = S^2 I_r = \begin{bmatrix} \frac{1}{\alpha_1} & & & \\ & \dots & & \\ & & \frac{1}{\alpha_r} & \\ & & & 0 \\ & & & & 0 \\ & & & & & 0 \end{bmatrix}$$

and transform back, $P^T(SI_r S)P$.

The same procedure works for a real symmetric matrix A whether positive semidefinite or not. The only shift in thought is that S is complex. But $SI_r S = S^2 I_r$ is real, so A^+ has the real factors $P^T(SI_r S)P$. Our conclusions are summarized in:

Theorem 1. To find the generalized inverse of a real symmetric matrix A find an orthogonal matrix P such that

$$PAP^T = \begin{bmatrix} \alpha_1 & & & \\ & \dots & & \\ & & \alpha_r & \\ & & & 0 \\ & & & & \dots \\ & & & & & 0 \end{bmatrix}$$

The rows of P are the eigenvectors of A and the diagonal values $\alpha_1, \dots, \alpha_r, 0, \dots, 0$ are the corresponding eigenvalues. The pseudo-inverse A^+ of A is

$$A^+ = P^T \begin{bmatrix} \frac{1}{\alpha_1} & & & \\ & \dots & & \\ & & \frac{1}{\alpha_r} & \\ & & & 0 \\ & & & & \dots \\ & & & & & 0 \end{bmatrix} P$$

Theorem 2. If a matrix A is of the form

$$\begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} \text{ where } B \text{ is non-singular,}$$

then

$$A^+ = \begin{pmatrix} B^{-1} & 0 \\ 0 & 0 \end{pmatrix}$$

Proof: The four properties can be shown to hold by matrix multiplication applied to submatrices.

Theorem 3. If $B = PAP^T$, where P is orthogonal, then $B^+ = PA^+P^T$.

Proof. To prove: $BPA^+P^TB = B$

$$\text{i.e., } P^TBPA^+P^TBP = P^TBP$$

$$\text{i.e., } A A^+ A = A \text{ which it is}$$

because $A = P^TBP$. Property 1 proved.

$$\text{To prove: } PA^+P^TBPA^+P^T = PA^+P^T$$

which it is because P^TBP in the middle = A and $A^+AA^+ = A^+$. Property 2 proved.

$$\text{To prove: } (BPA^+P^T)^T = BPA^+P^T$$

$$\text{i.e., } PA^+P^TB^T = BPA^+P^T$$

$$\text{i.e., } P^TPA^+P^TB^TP = P^TBPA^+P^TP$$

$$\text{i.e., } A^+A^T = AA^+ \text{ because } P^TP = I \text{ and } P^TB^TP = A^T$$

$$\text{i.e., } (AA^+)^T = AA^+ \text{ which it is}$$

because A^+ has Property 3. Property 3 proved. The proof of Property 4 is analogous.

2. A METRIC FOR PARTITIONING THE ANCILLARY VARIABLE SPACE WHEN THE COVARIANCE MATRIX IS SINGULAR

We suppose, as in the previous appendix that the spectral observations y and the ancillary variables v jointly have a covariance matrix

$$C_V^y = \begin{pmatrix} C_y & C_{yv} \\ C_{yv}^T & C_v \end{pmatrix}$$

and that the $m \times m$ covariance matrix C_v is singular, having rank $r < m$. Then there exists an $m \times m$ orthogonal matrix P such that

$$PC_vP^T = \begin{bmatrix} \alpha_1 & & & \\ & \dots & & \\ & & \alpha_r & \\ & & & 0 & \dots & 0 \end{bmatrix} = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}$$

where D is diagonal. The rows of P are the eigenvectors of C_v and $\alpha_1, \dots, \alpha_r, 0, \dots, 0$, the corresponding eigenvalues. We get the eigenvalues in the upper left, for convenience of notation, by rearranging the rows of P if necessary. This does not destroy the orthogonality of P .

Let

$$w = Pv$$

$$\text{cov } w = PC_vP^T = \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix}$$

$w_{r+1}, \dots, w_m$ are constant because they have zero variances.

$$\begin{pmatrix} y \\ w \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & P \end{pmatrix} \begin{pmatrix} y \\ v \end{pmatrix} \stackrel{\text{def}}{=} R \begin{pmatrix} y \\ v \end{pmatrix}$$

$$\text{cov} \begin{pmatrix} y \\ w \end{pmatrix} \stackrel{\text{def}}{=} C_w^y = \begin{pmatrix} I & 0 \\ 0 & P \end{pmatrix} \begin{pmatrix} C_y & C_{yv} \\ C_{yv}^T & C_v \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & P^T \end{pmatrix} \quad (1)$$

$$= \begin{pmatrix} C_y & C_{yv} \\ PC_{yv}^T & PC_v^T \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & P^T \end{pmatrix} = \begin{pmatrix} C_y & C_{yv}P^T \\ PC_{yv}^T & PC_vP^T \end{pmatrix}$$

In this matrix of submatrices,

$$PC_v P^T = \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix}$$

Now if w_i is constant, $\text{cov}(w_i, y_i) = 0$ because

$$\begin{aligned} \text{cov}(w_i, y_i) &= E w_i y_i - E w_i E y_i \\ &= w_i E y_i - w_i E y_i = 0 \end{aligned}$$

Hence, the bottom $m-r$ rows of PC_{yv}^T are 0 and the rightmost $m-r$ columns of $C_{yv} P^T$ are 0. So

$$\text{cov} \begin{pmatrix} y \\ w \end{pmatrix} = \begin{pmatrix} C_y & C_{yv} P_1^T & 0 \\ P_1 C_{yv}^T & D & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

where P_1 consists of the first r rows of P .

We define

$$C_v^{y+} = \begin{pmatrix} Q_y & Q_{yv} \\ Q_{yv}^T & Q_v \end{pmatrix}$$

as in Appendix III, except that the inverse is now the generalized inverse. Because

$$C_w^y = RC_v^{y+} R^T,$$

it follows from Theorem 3 that

$$\begin{aligned} C_w^{y+} &= RC_v^{y+} R^T \\ &= \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} Q_y & Q_{yv} \\ Q_{yv}^T & Q_v \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} Q_y & Q_{yv} P^T \\ P Q_{yv}^T & P Q_v P^T \end{pmatrix} \quad (2) \end{aligned}$$

after algebra like that of equation (1). By Theorem 2, since C_w^y is bordered with zeroes, so is C_w^{y+} .

Hence,

$$C_w^{y+} = \begin{bmatrix} Q_y & Q_{yv} P_1^T & 0 \\ P_1 Q_{yv}^T & E & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3)$$

In Appendix III, we defined the metric in ancillary variable space as a covariance matrix

$$\Sigma_v = (Q_{yv}^T Q_y^{-1} Q_{yv})^{-1}$$

In a clustering operation, we would assign a point v to a cluster with mean v_0 if the distance

$$(v - v_0)^T \Sigma_v^{-1} (v - v_0)$$

were a minimum for all cluster means. Thus, it is

$$\Sigma_v^{-1} = Q_{yv}^T Q_y^{-1} Q_{yv} \stackrel{\text{def}}{=} M_v$$

that would define the metric used for clustering if C_v^y were non-singular. From Equation (2) we would similarly define the clustering metric in w space as

$$M_w = P Q_{yv}^T Q_y^{-1} Q_{yv} P^T$$

From Equations (2) and (3) we infer that

$$P Q_{yv}^T = \begin{pmatrix} P_1 \\ 0 \end{pmatrix} Q_{yv}^T$$

$$Q_{yv} P^T = Q_{yv}^T (P_1 \ 0)$$

Hence,

$$M_w = \begin{pmatrix} P_1 Q_{yv}^T Q_y^{-1} Q_{yv} P_1^T & 0 \\ 0 & 0 \end{pmatrix} \quad (4)$$

We note, in passing, that because M_w has rank r , $M_v = P^T M_w P$ has rank r .

The result of Appendix III holds only for a non-singular covariance matrix of ancillary variables. Hence, we apply it to the set of variables $w_1, \dots, w_r$ and obtain the metric

$$P_1 Q_{yv}^T Q_y^{-1} Q_{yr} P_1^T$$

When this metric is embedded in the full w space, it becomes M_w as given in Equation (4). The last $m-r$ values of w are thus provided for, but they do not affect the distance calculation. The distance from a w point to a cluster mean w_o in w space is

$$\begin{aligned} & (w - w_o)^T M_w (w - w_o) \\ &= (w - w_o)^T (P Q_{yv}^T Q_y^{-1} Q_{yv} P^T) (w - w_o) \\ &= (P^T w - P^T w_o)^T (Q_{yv}^T Q_y^{-1} Q_{yv}) (P^T w - P^T w_o) \end{aligned}$$

Now $w = Pv$ so $v = P^T w$. Let v and v_o be the same setting of ancillary variables as w and w_o but expressed in v coordinates. The distance just defined is

$$(v - v_o)^T (Q_{yv}^T Q_y^{-1} Q_{yv}) (v - v_o)$$

Thus, $M_v = Q_{yv}^T Q_y^{-1} Q_{yv}$ is the metric to be used for clustering in v space.

3. CONCLUSIONS

The lengthy discussion in this appendix results in a simply-stated conclusion, as follows. Let

$$\begin{pmatrix} Q_y & Q_{yv} \\ Q_{yv}^T & Q_v \end{pmatrix}$$

be the inverse of

$$\text{cov} \begin{pmatrix} y \\ v \end{pmatrix},$$

if it has one, and the generalized inverse otherwise. The metric for clustering in ancillary variable space is

$$Q_{yv}^T Q_y^{-1} Q_{yv}$$

This metric is the same as the one derived in Appendix III except that it is more usefully defined as the matrix M in the distance

$$(v - v_0)^T M (v - v_0)$$

rather than as $M^{-1} = \Sigma_V$.

To find the generalized inverse C_V^{y+} of $\text{cov}(Y)$ find the matrix P of eigenvectors (by rows) and the corresponding eigenvalues $\alpha_1, \dots, \alpha_{n+r}, 0, \dots, 0$.

$$C_V^{y+} = P^T \begin{bmatrix} \frac{1}{\alpha_1} & & & \\ & \dots & & \\ & & \frac{1}{\alpha_r} & \\ & & & \dots \\ & & & & 0 \end{bmatrix} P$$

APPENDIX III.2

INVARIANCE OF THE CONDITIONAL COVARIANCE

(W. Holsztynski)

1. INTRODUCTION

The main goal of this note is to prove a conjecture in linear algebra of R. Kauth and W. Richardson. Suppose that we are given α - and γ -dimensional vectors

$$y_k = \begin{bmatrix} 1 \\ y_k^1 \\ \vdots \\ y_k^\alpha \end{bmatrix} \quad \text{and} \quad v_k = \begin{bmatrix} 1 \\ v_k^1 \\ \vdots \\ v_k^\gamma \end{bmatrix}, \quad \text{where } k=1, \dots, N. \quad (1)$$

In the context of multispectral data processing, the y and v vectors represent spectral and ancillary data vectors, respectively. We can introduce $(\alpha+\gamma)$ -dimensional vectors

$$z_k = \begin{bmatrix} y_k \\ v_k \end{bmatrix} = \begin{bmatrix} 1 \\ y_k^1 \\ \vdots \\ y_k^\alpha \\ 1 \\ v_k^1 \\ \vdots \\ v_k^\gamma \end{bmatrix} \quad \text{for } k=1, \dots, N. \quad (2)$$

and matrices

$$Y = [\bar{y}_1, \dots, \bar{y}_N], \quad V = [\bar{v}_1, \dots, \bar{v}_N], \quad Z = \begin{bmatrix} Y \\ V \end{bmatrix} = [\bar{z}_1, \dots, \bar{z}_N], \quad (3)$$

where

$$\bar{y}_k = y_k - \bar{y}, \quad \bar{v}_k = v_k - \bar{v}, \quad \bar{z}_k = z_k - \bar{z} = \begin{bmatrix} \bar{y}_k \\ \bar{v}_k \end{bmatrix} \quad (4)$$

and $\bar{y}$, $\bar{v}$ and $\bar{z}$ are the respective means:

$$\bar{y} = \frac{1}{N}(y_1 + \dots + y_N), \quad \bar{v} = \frac{1}{N}(v_1 + \dots + v_N), \quad \bar{z} = \begin{bmatrix} \bar{y} \\ \bar{v} \end{bmatrix} = \frac{1}{N}(z_1 + \dots + z_N). \quad (5)$$

Suppose that the rank of Z is $\alpha + \gamma$, i.e., that the covariance matrix of Z

$$\sum = \frac{1}{N} ZZ^t \quad (6)$$

is a non-singular matrix (it follows that

$$\sum_{yy} = \frac{1}{N} YY^t \quad \text{and} \quad \sum_{vv} = \frac{1}{N} VV^t \quad (7)$$

are also non-singular). Then we may consider the multivariate normal probability density

$$d(z) = k e^{-\frac{1}{2} z^t \sum^{-1} z}, \quad \text{where } z = \begin{bmatrix} z_1 \\ \vdots \\ z_{\alpha+\gamma} \end{bmatrix} \in R^{\alpha+\gamma} \quad (8)$$

One can also compute the conditional density function $d_z(y|v)$, where

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_\alpha \end{bmatrix} \in R^\alpha \quad \text{and} \quad v = \begin{bmatrix} v_1 \\ \vdots \\ v_\gamma \end{bmatrix} \in R^\gamma \quad (9)$$

In order to do it let us write

$$\Sigma^{-1} = \begin{bmatrix} A & B \\ B^t & C \end{bmatrix} \quad (10)$$

where A and C are $\alpha \times \alpha$ and $\gamma \times \gamma$ matrices respectively. Then

$$d(y|v) = \exp \left\{ -\frac{1}{2}(y + A^{-1}Bv)^t A(y + A^{-1}Bv) \right\} \quad (11)$$

by Equation 6 of Appendix III.1 or by Section 4 of this appendix. Observe that A^{-1} is the covariance matrix of $y|v$ and $-A^{-1}Bv$ is the mean. By substituting $\Omega \Omega^t$ for A and u for $\Omega^{-1}Bv$, we may write the above formula in a compact way:

$$d(y|v) = \exp \left\{ -\frac{1}{2}(\Omega^t y + u)^t (\Omega^t y + u) \right\} \quad (12)$$

Because of this and for some other reasons R. Kauth decided to replace vectors $v_1, \dots, v_N$ by $u_1 = \Omega^{-1}Bv_1, \dots, u_N = \Omega^{-1}Bv_N$, and he and W. Richardson ran the above computations again in order to compute the new covariance matrix A_U , of the conditional density $d(y|u)$ (corresponding to the old covariance matrix of (11) above), and the new mean value $A_U^{-1}B_U u$, of $d(y|u)$ (the old one, given by (11), was $A^{-1}Bv$). They were surprised to find that

$$A_U = A \quad \text{and} \quad A_U^{-1}B_U u = A^{-1}Bv, \quad (13)$$

i.e., the conditional covariance matrix and the conditional mean value were invariant under the linear map

$$L = \Omega^{-1}B. \quad (14)$$

Since the concrete matrix Z they started with was sizeable and without any pattern, they conjectured that (13) holds always, granted that the covariance matrix of u , $\sum_U = \frac{1}{N} Z_U Z_U^t$, is invertible; here

$$Z_U = \begin{bmatrix} Y \\ U \end{bmatrix}, \quad U = [u_1, \dots, u_N]. \quad (15)$$

The invertibility of the covariance matrix of Z_U is discussed in Section 2. In Section 3 the proof of the conjecture is given. Section 4 contains an alternate derivation of the conditional density of $Y|V$.

2. INVERTIBILITY OF THE COVARIANCE OF Z_U

We keep the same notation as in Section 1. In particular:

$$\begin{aligned} A &= \Omega \Omega^t \\ L &= \Omega^{-1} B \end{aligned} \quad (16)$$

$$u_k = L v_k \quad \text{for } k=1, \dots, N; \quad (17)$$

$$U = [u_1, \dots, u_N], \quad Z_U = \begin{bmatrix} Y \\ U \end{bmatrix}. \quad (18)$$

We want to know when

$$\sum_U = \frac{1}{N} Z_U Z_U^t \quad (19)$$

is a non-singular matrix. It is known that this happens if and only if the rank of Z_U is 2α , i.e., when there are 2α linearly independent vectors among

$$\begin{bmatrix} y_1 \\ u_1 \end{bmatrix}, \dots, \begin{bmatrix} y_N \\ u_N \end{bmatrix}$$

(observe that u_k are α -dimensional vectors). But we do not want a condition which involves all $u_1, \dots, u_N$.

Remark 1

In what follows, in this section and in Section 3, the results are valid even if Y , V and Z do not have mean 0 (in other words $\bar{y}_k, \bar{v}_k, \bar{z}_k$ of (3) can be replaced by y_k, v_k, z_k).

Theorem 1

$\sum_U$ is non-singular if and only if $\text{rank } B = \alpha$. (We always assume that $\sum = \frac{1}{N} ZZ^t$ is non-singular in first place!)

Proof

Since $\sum$ is non-singular there are indices

$$1 \leq i_1 < \dots < i_\gamma \leq N \quad \text{and} \quad 1 \leq j_1 < \dots < j_\alpha \leq N$$

such that $v_{i_1}, \dots, v_{i_\gamma}$ are linearly independent in R^γ , $y_{j_1}, \dots, y_{j_\alpha}$ are linearly independent in R^α , and $i_n \neq j_m$ for any $n=1, \dots, \gamma$ and $m=1, \dots, \alpha$. To justify this statement, we can choose $\alpha+\gamma$ linearly independent z vectors from among the N given and consider them as a matrix, each column of which is a z vector. The top α values of each column is a y vector and the bottom γ values, a v vector. The determinant of the matrix can be expressed as a sum of terms, each of which is a minor determinant in the top half times a complementary minor determinant in the bottom half. Because the determinant of the matrix is not zero, there is a pair of complementary minors whose determinants are not zero. An acceptable set of indices $j_1, \dots, j_\alpha$ corresponds to the columns of the top minor and a disjoint set $i_1, \dots, i_\gamma$ corresponds to the columns of the bottom minor. Thus the proof of the statement is established.

If rank $B = \alpha$ then there are α linearly independent vectors among

$$u_{i_1} = \Omega^{-1} B v_{i_1}, \dots, u_{i_\gamma} = \Omega^{-1} B v_{i_\gamma}$$

(because when a matrix D is non-singular, rank of $DC = \text{rank of } C$).

This means that there are 2α linearly independent vectors among $\alpha + \gamma$ vectors

$$\begin{bmatrix} y_{i_1} \\ u_{i_1} \end{bmatrix}, \dots, \begin{bmatrix} y_{i_\gamma} \\ u_{i_\gamma} \end{bmatrix}, \begin{bmatrix} y_{j_1} \\ u_{j_1} \end{bmatrix}, \dots, \begin{bmatrix} y_{j_\alpha} \\ u_{j_\alpha} \end{bmatrix}.$$

Thus $\sum_U$ is non-singular.

Conversely, if $\sum_U$ is non-singular then there are α , and not more than α linearly independent vectors among $u_1, \dots, u_N$. Thus rank $B = \text{rank } L = \alpha$. Theorem is proved.

Remark 2

The obvious necessary condition for $\sum_U$ to be non-singular is $\gamma \geq \alpha$. The class of examples below show that this condition is not sufficient.

Example 1

Let linearly independent vectors

$$a^k = [a_1^k, \dots, a_N^k] \in R^N \quad \text{for } k = 1, \dots, \alpha + \gamma$$

be such that a^k is orthogonal to a^ℓ whenever

$$1 \leq k \leq \alpha < \ell \leq \alpha + \gamma.$$

Next, let

$$y_i = \begin{bmatrix} 1 \\ a_i \\ \vdots \\ a_i^\alpha \end{bmatrix} \quad \text{and} \quad v_i = \begin{bmatrix} \alpha+1 \\ a_i \\ \vdots \\ a_i^{\alpha+\gamma} \end{bmatrix} \quad \text{for } i = 1, \dots, N.$$

Then $\text{rank } Z = \alpha + \gamma$. Thus Σ is non-singular. On the other hand $YV^t = 0$, i.e.,

$$\Sigma = \frac{1}{N} \begin{bmatrix} YY^t & 0 \\ 0 & VV^t \end{bmatrix}$$

It follows that Σ^{-1} is of the form

$$\Sigma^{-1} = N \begin{bmatrix} (YY^t)^{-1} & 0 \\ 0 & (VV^t)^{-1} \end{bmatrix}$$

Thus $B = 0$ and $\text{rank } B = 0$. Thus, by the above theorem, Σ_U is singular.

The simplest example of this type is:

$$\alpha = \gamma = 1,$$

$$y_1 = 1 \quad y_2 = 0$$

$$v_1 = 0 \quad v_2 = 1.$$

Then

$$2\bar{\Sigma} = \bar{Z}\bar{Z}^t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \frac{1}{2} \bar{\Sigma}^{-1}$$

and $B = 0$.

The simplest example with the averages $\bar{y}$, $\bar{v}$ (and $\bar{z}$) equal 0 is:

$$\alpha = \gamma = 1,$$

$$\begin{array}{lll} y_1 = 1 & y_2 = -1 & y_3 = 0 \\ v_1 = 1 & v_2 = 1 & v_3 = -2; \end{array}$$

and again $B = 0$.

Remark 3

Rank $B = \text{rank } L = \text{rank } A^{-1}B$.

3. PROOF OF THE CONJECTURE

In this section we assume that both

$$\bar{\Sigma} \text{ and } \bar{\Sigma}_U = \begin{bmatrix} Y \\ U \end{bmatrix} [Y^t U^t]$$

are non-singular.

Theorem 2

$$\bar{\Sigma}_U^{-1} = \begin{bmatrix} A & \Omega \\ \Omega^t & I + N(UU^t)^{-1} \end{bmatrix}.$$

Proof

Since $\sum^{-1} = I$ hence

$$A Y Y^t + B V Y^t = N I \quad (20)$$

and

$$A Y V^t + B V V^t = 0. \quad (21)$$

Indeed, we have

$$\begin{bmatrix} A & B \\ B^t & C \end{bmatrix} \begin{bmatrix} Y V^t & Y V^t \\ V Y^t & V V^t \end{bmatrix} = N \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}.$$

Now, put

$$\begin{bmatrix} A & \Omega \\ \Omega^t & I + N(UU^t)^{-1} \end{bmatrix} \begin{bmatrix} Y Y^t & Y U^t \\ U Y^t & U U^t \end{bmatrix} = \begin{bmatrix} P & R \\ Q & S \end{bmatrix}, \quad (22)$$

where P and S are $\alpha \times \alpha$ and $\gamma \times \gamma$ matrices respectively. Then

$$\begin{aligned} P &= A Y Y^t + \Omega U Y^t = A Y Y^t + \Omega (\Omega^{-1} B V) Y^t \\ &= A Y Y^t + B V Y^t = N I \quad (\text{by (20)}) \end{aligned} \quad (23)$$

and

$$\begin{aligned} R &= A Y U^t + \Omega U U^t = A Y V^t (\Omega^{-1} B)^t + \Omega (\Omega^{-1} B V) V^t (\Omega^{-1} B)^t \\ &= (A Y V^t + B V V^t) (\Omega^{-1} B)^t = 0 \quad (\text{by (21)}). \end{aligned} \quad (24)$$

Since $R=0$ hence

$$AYU^t = -\Omega UU^t$$

and substituting $\Omega\Omega^t$ for A ,

$$-\Omega^{-1} = (UU^t)^{-1}UY^t.$$

Now we can compute Q ,

$$\begin{aligned} Q &= \Omega^t Y Y^t + (I + N(UU^t)^{-1})UY^t \\ &= \Omega^{-1}(A Y Y^t + \Omega U Y^t) + N(UU^t)^{-1}UY^t \\ &= N\Omega^{-1} - N\Omega^{-1} = 0 \quad (\text{by (23)}). \end{aligned} \tag{25}$$

Also, by (22) and (23),

$$\begin{aligned} S &= \Omega^t Y U^t + (I + N(UU^t)^{-1})UU^t \\ &= \Omega^{-1}(A Y U^t + \Omega U U^t) + NI = NI. \end{aligned} \tag{26}$$

The theorem is proved.

The above proof can have a more conceptual finish. After formula (24) was displayed we could see that $\sum_U^{-1}$ must be of the form

$$\sum_U^{-1} = \begin{pmatrix} A & \Omega \\ F & G \end{pmatrix},$$

where F and G are some $n \times n$ matrices. Since $\sum_U$ is symmetric, so is $\sum_U^{-1}$. Thus

$$F = \Omega^t.$$

Finally, since

$$\begin{aligned} NI &= FYU^t + GUU^t = \Omega^t YU^t + UU^t + (G - I)UU^t \\ &= \Omega^{-1}(AYU^t + \Omega UU^t) + (G - I)UU^t = (G - I)UU^t \end{aligned}$$

hence

$$G = I + N(UU^t)^{-1}.$$

The theorem is proved again.

Corollary 1

Let us consider the multivariate normal distribution which has

$$d_U(z) = ce^{-\frac{1}{2} z^t \Sigma_U^{-1} z} \quad (z \in R^{2\alpha}),$$

as the density function. Put

$$z = \begin{bmatrix} y \\ u \end{bmatrix}, \quad \text{where } y, u \in R^\alpha.$$

Then

$$d_U(y|u) = pe^{-\frac{1}{2} (y + A^{-1}\Omega u)^t A (y + A^{-1}\Omega u)} \quad (\text{as in (11)}).$$

Because $u = \Omega^{-1}Bv$,

$$A^{-1}\Omega u = A^{-1}Bv$$

Thus $d(y|v)$ and $d_U(y|u)$ have the same mean and covariance matrix, proving the conjecture. We note further that

$$A^{-1}\Omega u = (\Omega\Omega^t)^{-1}\Omega u = \Omega^{-1}u$$

so $d_U(y|v)$ can also be written in the form

$$d_U(y|u) = \rho e^{-\frac{1}{2} (y + \Omega^{-1}u)^t A (y + \Omega^{-1}u)}$$

Corollary 2

The $\alpha\alpha$ matrix YU^t is non-singular.

Proof

By (24),

$$AYU^t + \Omega UU^t = 0.$$

Thus

$$YU^t = -\Omega^{-1}UU^t,$$

which is non-singular because UU^t is a principal minor of the positive definite matrix $N_{\sum U}^t$.

Remark

Put $L_v = L = \Omega^{-1}B$. If we apply the same operation again, then $A = A_v$ should be replaced by A_u which happens to be A again. And $B_v = B$ should be replaced by $B_u = \Omega$. Thus we see that the second iteration of Kauth's operation

$$L_u = \Omega_u^{-1}B_u = \Omega^{-1}\Omega = I$$

is already identity. The process stabilizes after the first step.

4. ALTERNATE DERIVATION OF THE CONDITIONAL DENSITY OF $y|v$

Let M be an arbitrary positive definite symmetric $(\alpha+\gamma) \times (\alpha+\gamma)$ matrix. We can write it in the form

$$M = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \quad (27)$$

where A and C are $\alpha \times \alpha$ and $\gamma \times \gamma$ matrices respectively. Then A and C are positive definite and symmetric.

Consider the normal zero-mean density function

$$d(z) = k e^{-\frac{1}{2} z^T M z} \quad \text{for } z \in R^{\alpha+\gamma} \quad (28)$$

Write z as

$$z = \begin{bmatrix} y \\ v \end{bmatrix} \quad \text{where } y \in R^\alpha \text{ and } v \in R^\gamma. \quad (29)$$

We want to compute $d(y|v)$. Observe that

$$\begin{aligned} z^T M z &= \begin{bmatrix} y \\ v \end{bmatrix}^T \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \begin{bmatrix} y \\ v \end{bmatrix} \\ &= y^T A y + v^T B^T y + y^T B v + v^T C v \\ &= (y + A^{-1} B v)^T A (y + A^{-1} B v) + v^T (C - B^T A^{-1} B) v. \end{aligned} \quad (30)$$

To compute the marginal density of v , we consider the integral

$$\lambda = \int_{R^{\alpha}} \kappa e^{-\frac{1}{2} (y + A^{-1}Bv)^t A (y + A^{-1}Bv)} dy \quad (31)$$

Except for the constant κ , it has the form of an integral of a multivariate normal distribution over R^{α} , the space relevant to y . It is thus invariant to the choice $A^{-1}Bv$ of the mean and y integrates out. Thus λ is a constant depending only on A (but not on v , y , or B).

Thus the marginal density of v is

$$D(v) = \lambda e^{-\frac{1}{2} v^t (C - B^t A^{-1} B) v} \quad (32)$$

The conditional density

$$d(y|v) = \frac{d(z)}{D(v)} \quad (33)$$

is given now by

$$d(y|v) = \kappa e^{-\frac{1}{2} (y + A^{-1}Bv)^t A (y + A^{-1}Bv)} \quad (34)$$

We conclude that the conditional density $d(y|v)$ is normal; A^{-1} is its conditional covariance and $-A^{-1}Bv$ is its mean value.

APPENDIX IV

LAST MINUTE RESULTS AND AN EXPLANATION OF THE SIGNATURE EXTENSION PROBLEM

This appendix is written and added in final proof. We feel that the information contained here will add considerably to the understanding of the power and limitations of Procedure B.

We have satisfied ourselves that a major source of error is simply the use of a large partition. We were led to this conclusion by finding that some large spectral strata which are nearly pure wheat in a majority of the sample segments are non-wheat in a certain small subset of the segments resulting in large overestimates (Segments 1163, 1165 and 1167). Spectral strata which are nearly pure non-wheat for a majority of the segments have significant wheat percentages for a different small subset, resulting in sizeable underestimates (Segments 1861 and 1865). These examples are shown in Figure 1, which is a portion of a table of true percent wheat by spectral stratum and segment, and in Figure 2, which is a reproduction of Figure 10 (Section 5) showing key segment numbers.

We have run the spectral stratification program, BCLUST, with a variety of parameter settings and using a variety of distance measures. Emphasis has been on using a large number of small spectral strata, and the above statements are verified in each case. Hence we conclude that the non-wheat in Segments 1163, 1165 and 1167, for example, cannot be separated from the wheat in the majority of the segments on the basis of spectral information alone. Some other information, either as input to a signature model or as a partitioning variable, must be used.

The most noticeable fact about the segments identified in Figure 2 is their position in the State of Kansas. Figure 3 shows these same segments and their relationship to others in the State. The association of the two figures is obvious. Hence we sought supporting information in other ancillary information besides geographical position.

of the segments are represented by two different combinations of passes.)

SPECTRAL STRATUM

FORNERS OF AFRICAN TRIBE - **NOT** INDEPENDENT

ORIGINAL PAGE IS
OF POOR QUALITY

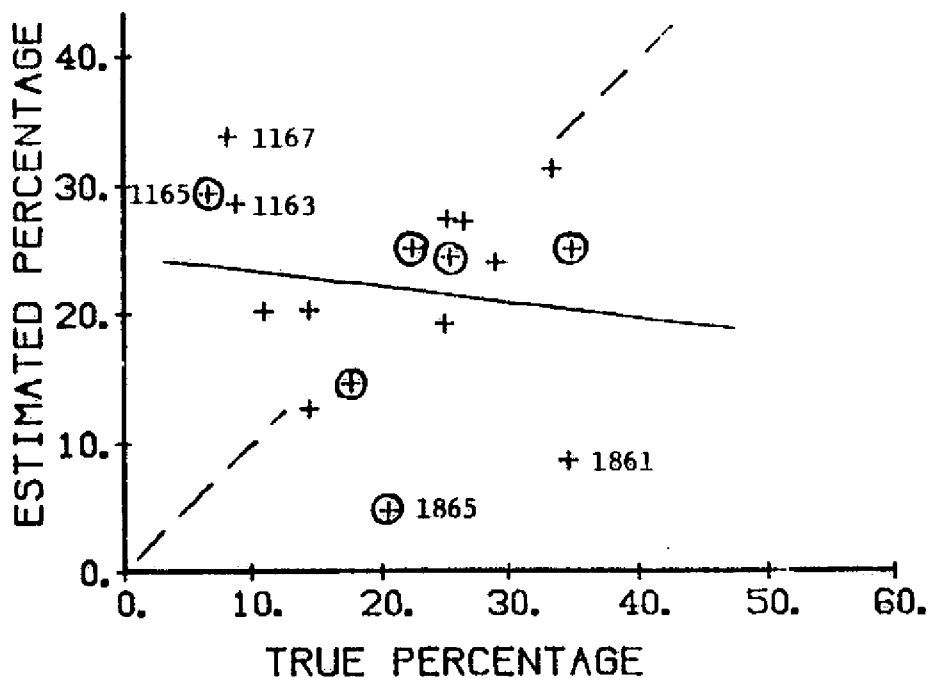


FIGURE 2. REPRODUCTION OF FIGURE 10 (SECTION 5),
SHOWING KEY SEGMENT NUMBERS

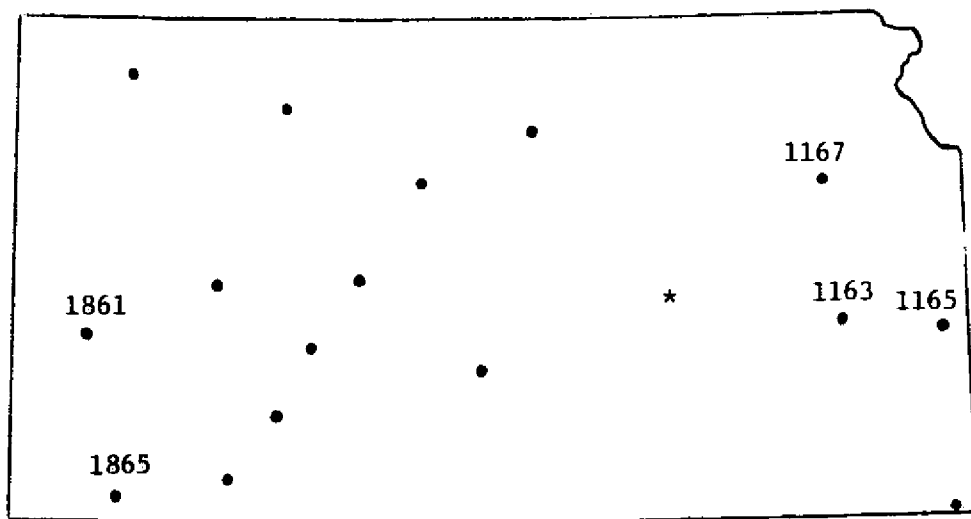


FIGURE 3. LOCATION IN KANSAS OF THE SEGMENTS STUDIED,
SHOWING KEY SEGMENT NUMBERS

We first considered the "crop calendar", as illustrated by the Robertson Triquadratic Model to compute biometeorological time, output from CCEA(NOAA). These data were provided as part of the LACIE Yield Estimation Subsystem Weekly Meteorological Summaries. An example is shown in Figure 4 which gives the crop calendar adjustment for April 18, 1976. Although there is a slight east-west trend, the dominant change is from north to south. It appears doubtful that the differences in signature are explained by this source. In general, the crop calendar is defined by the daily maximum and minimum temperatures, and in general temperature gradients are maximum in the north-south direction. Furthermore, the average crop calendar is partially compensated already due to the fact that LACIE acquisition windows are defined by the average crop calendar.

Next we considered the various precipitation indices available from the Weekly Summaries, as well as the general reports of crop progress and condition. As is well known in the LACIE community, the 1975-76 season was a drought year in many parts of the Great Plains. In the State of Kansas we find an easily observable correlation with crop moisture indices. Figure 5 is a typical early season crop moisture index. The eastern part of KANSAS had received adequate moisture -- the southwestern corner was "too dry, yield prospects reduced". The western half was "abnormally dry, prospects deteriorating".

This overall pattern was maintained from September 1975 through February 1977. Associated lack of snow cover exposed the west and southwest portions to winter kill. By March farmers were abandoning fields.

The National Almanac shows the long term average precipitation index to have essentially east-west gradients across Kansas. In 1975-76 this gradient was unusually strong, with emphasis on drought in the southwest corner of the state.

ORIGINAL PAGE IS
OF POOR QUALITY

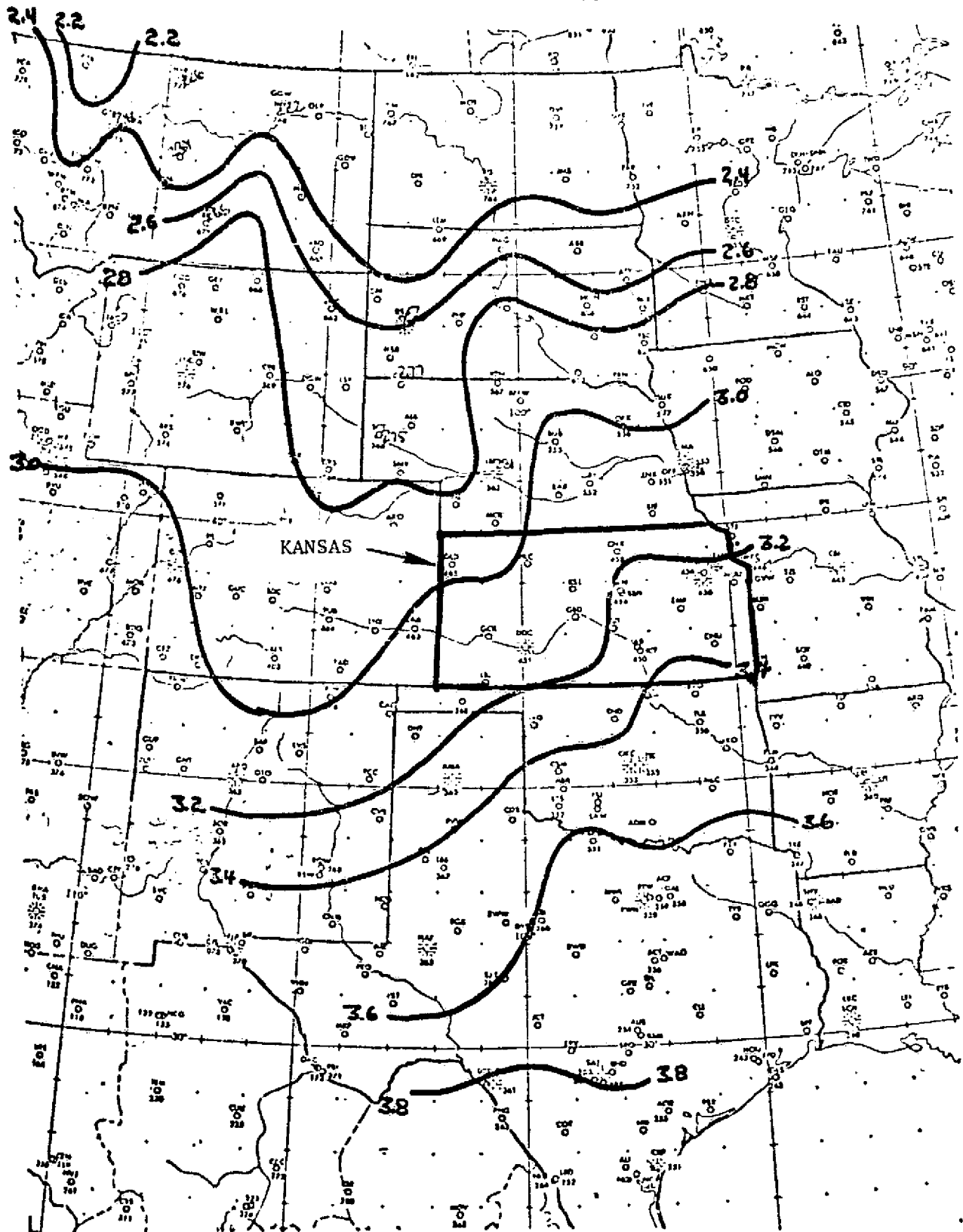
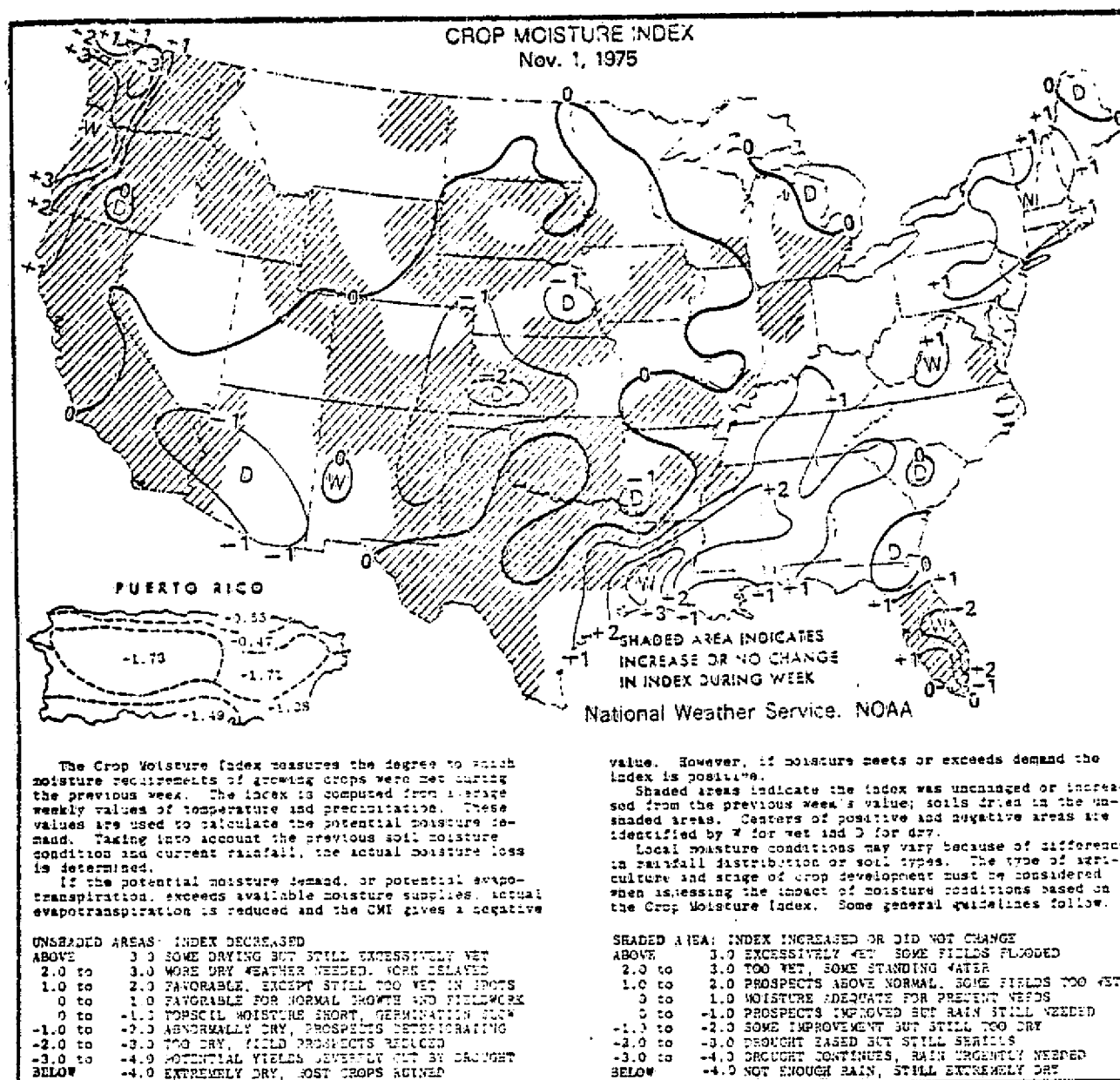


FIGURE 4. MAP OF CROP CALENDAR ADJUSTMENTS (APRIL 18, 1976)

FIGURE 5. TYPICAL EARLY-SEASON MAP OF THE CROP MOISTURE INDEX



One might expect that the development of the crop would be delayed but this was not the case. In fact an early warming trend brought winter wheat fields out of dormancy earlier than usual throughout the state, providing they had survived the drought. The crop development stayed ahead of normal throughout the season. The principal effect of the drought upon surviving crops appears to have been to reduce the ground cover of the crop.

We can see in this pattern of events a rationale for the pattern of missed wheat detections evident in Figures 2 and 3. The "greenness" of the wheat signature is a function primarily of ground cover, given that the leaves themselves are green. Grass or pasture contains a great deal of dead material, so that the new shoots are in part hidden and the potential for green cover is initially small, compared to that for wheat.

In this case, however, the wheat signature is in fact dominated by the large dry central region of the state (even though, out of the six training segments, one each is chosen from the east and the southwest) and this signature matches grass/pasture in the eastern part of the state. In the southwest the wheat fields are of such low ground cover that they look like grass/pasture in the central section.

The way to test this conjecture is to incorporate crop moisture index as an ancillary variable to be used either to partition the data set or to make the signatures dependent upon it. At the present time we do not have these data available in a suitable form. However the overall trend appears to be that the average gradient is lined up with the progression of longitudes. Hence we have elected to use segment longitude as an ancillary variable in a preliminary attempt to improve results.

The method we have chosen is to incorporate longitude along with spectral variables into the stratification program, BCLUST. The result is to effect a joint spectral and spatial stratification. We have attempted this approach previously using linear combinations of many

ancillary variables and have found that great care is needed to keep the ancillary variables from dominating the stratification.

Figure 6 shows the result of applying this procedure. These estimates are notably improved over previous ones. The bulk of the observations are near the 45° diagonal (dashed line) and in particular the observations which previously were wild are close to the line. An overall slightly positive bias is evident. There is one segment, 1883, identified as "*" in Figure 3, which is greatly overestimated. In the joint spectral-spatial stratification this segment formed a stratum unto itself; but unfortunately the ground truth had not been exhaustively prepared for this segment so it could not be used for training, hence this wild estimate.

One is tempted to believe that the east-west pattern (actually a surrogate for the east-west moisture pattern) explains the Procedure B performance and signature differences satisfactorily. However the results are compromised by the fact that we used the true proportions from all the sites in looking for an explanation. For this reason we are not stating the results as a conclusive test. Instead we express the explanation for the results in terms of a hypothesis to be tested; namely, winter wheat spectral signatures, especially the "greenness" component, are more significantly related to ground cover and thus to available soil moisture, than they are to departures of temperature conditions from the long term average.

An empirical study of spectral-temporal signature patterns is needed to further reinforce the viewpoint developed here. We expect to carry out such studies along with other ancillary data studies in the future.

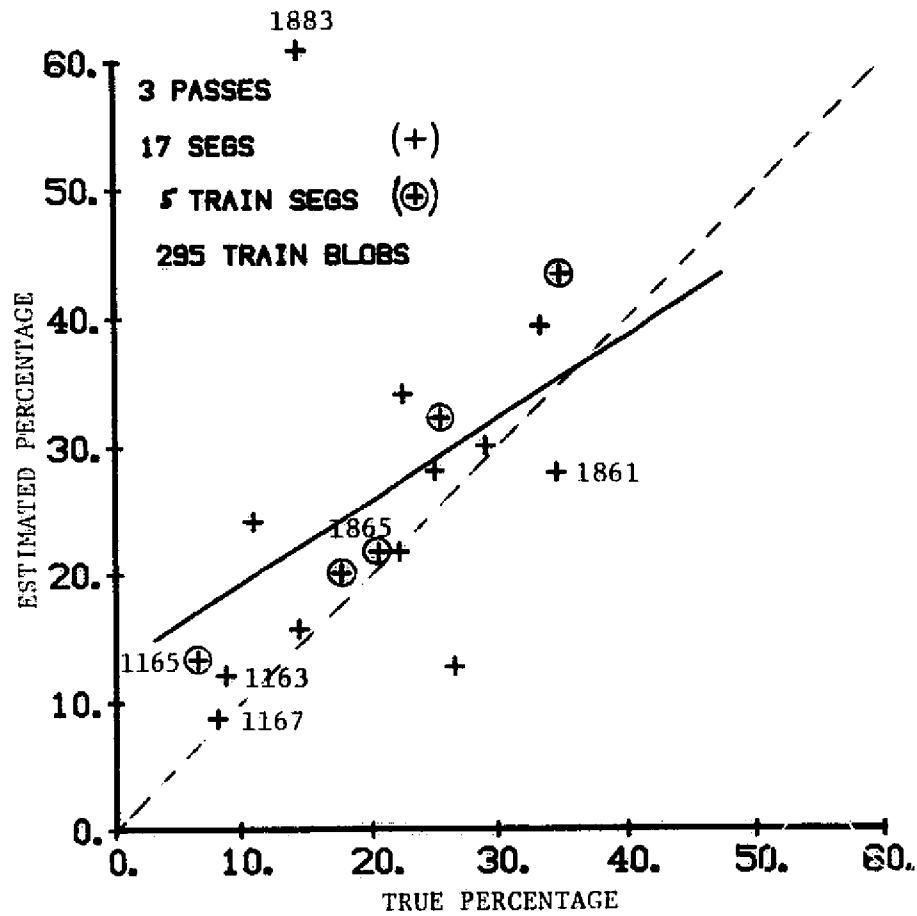


FIGURE 6. ESTIMATED VS. TRUE WHEAT PERCENTAGE FOR 17 SEGMENTS WHEN LONGITUDE IS INCLUDED IN THE STRATIFICATION

REFERENCES

1. Kauth, R. J. and G. S. Thomas, "The Tasselled Cap -- A Graphic Description of the Spectral-Temporal Development of Agricultural Crops As Seen by Landsat", Symposium on Machine Processing of Remotely Sensed Data June 29 - July 1, 1976, Laboratory for Applications of Remote Sensing, Purdue University, W. Lafayette, Indiana, 1976.
2. Malila, W. A., R. B. Crane, and R. E. Turner, Information Extraction Techniques for Multispectral Scanner Data, Technical Report 31650-74-T, Willow Run Laboratories, The University of Michigan, Ann Arbor, Michigan, June 1972.
3. Kauth, R. J. and G. S. Thomas, System for Analysis of Landsat Agricultural Data, Technical Report 109600-67-F, Environmental Research Institute of Michigan, Ann Arbor, Michigan, May 1976.
4. Kauth, R. J., A. P. Pentland, and G. S. Thomas, "BLOB, An Unsupervised Clustering Approach to Spatial Preprocessing of MSS Imagery", Eleventh International Symposium on Remote Sensing of Environment April 25-29, 1977, Vol. 2, Environmental Research Institute of Michigan, Ann Arbor, 1977, pp. 1309-1317.
5. Lambeck, P. F., Signature Extension Preprocessing for Landsat MSS Data, Technical Report 122700-32-F, Environmental Research Institute of Michigan, Ann Arbor, Michigan, November 1977.
6. Odell, P. L. and T. L. Boullion, Generalized Inverse Matrices, New York, Wiley, 1971.
7. Rao, C. R. and S. K. Mitra, Generalized Inverse of Matrices and Its Applications, New York, Wiley, 1971.
8. Ben-Israel, A. and T. N. E. Greville, Generalized Inverses, Theory, and Applications, New York, Wiley, 1974.